

摘 要

近年来客运专线发展迅速,新的维修技术手段与先进检测设备的结合,大量检测数据、维修数据的分析和整合、各种维修作业管理的加速更新,需要新的现代化信息系统作为重要支撑。

论文首先通过对客运专线基础设施维修基地维修管理系统施工管理模块的需求情况进行分析,确定了系统的研究目标。以 JavaEE 技术为基础,首先分析了 MVC 模式和 Struts1.2、Beehive 两个框架。Beehive 是一个兼容 Struts1 的 MVC 框架。与 Struts1.2 中配置文件需要人工设置不同,Beehive 框架的页面流技术支持注释机制和元数据自动生成。从而自动生成配置文件。作为 Beehive 框架重要组成的控件子系统是一些被封装的业务逻辑组件,为所有的资源集提供一组通用的接口。降低了一个作为 J2EE 资源的客户机的复杂性。论文在 Beehive 框架的基础上进行分层设计,对业务逻辑的通用方法单独设计成业务基础层。提高了开发效率。然后引入 JBPM workflow 技术对各类施工计划上报审批进行详细分析与设计。JBPM workflow 是一个基于 JavaEE 的轻量级 workflow 系统,提供了流程定义、部署、执行和管理等功能。其次论文针对系统安全问题设计了权限管理方案,该方案在符合铁路业务需求基础上提取出通用标准数据接口,这种设计可以为各种业务应用系统提供统一的权限管理功能。在此基础上,以作者参与设计并开发的施工管理模块为例,在 Beehive 分层架构实现了施工管理业务功能,运用 JBPM workflow 引擎管理施工计划业务流程。

最后,详细分析设计了施工管理模块、施工计划流转子模块。具体分析和设计了系统多层结构、主要功能模块以及完成模块中关键技术的实现和系统的测试。目前,系统已经实际应用于上海、武汉等客运专线基础设施维修基地,实际应用表明,该系统实现了设计目标,性能可靠,赢得了用户的一致好评。

关键词: 客运专线; 施工管理; Beehive 架构; workflow; JavaEE

Abstract

Due to the rapid development of passenger dedicated lines, the combination of new maintenance methods and advanced examination equipment, and fast upgrade of all kinds of management, analysis and integration of a large amount of examination data and maintenance data, they all need great supports from new modernized information system.

This thesis firstly gives an analysis of the demand of construction management modules in maintenance management system of the PDL infrastructure, and determines the research target of the system. Based on JavaEE, it analyzes MVC module and two frameworks: Struts 1.2 and Beehive. Beehive is a MVC framework, which is compatible with Struts1. Page flow technology in Beehive framework supports the annotation and automatic metadata generation mechanism, and automatically generates configuration files, which is different from the manual set of Struts1.2configuration files. Some blocked business logic components play an important part as the control subsystem in the Beehive framework, which provide a set of generic interface for all resources and reduce complexity of J2EF resource as a client. This thesis conducts layered design based on Beehive framework, and separately designs business foundation for the general methods of business logic, which increases the development efficiency. Then it introduces JBPM work flow technology, and gives a detail analysis and design of all kinds of programs. JBPM is a work flow based on lightweight workflow system of JavaEE and provides various functions, such as process definition, deployment, implementation and management. Thirdly, it designs management scheme in the view of system safety. This scheme draws standard data interface based on the demand of suitable railway business and offers a variety of business application systems to provide a unified access management function. On the basis of this, taking the author's participation in the design and development of construction management module as an example, it realizes construction management business functions of Beehive layered structure and makes an application of JBPM work flow to manage the business.

Finally, this thesis gives a detail analysis of construction management module and construction plan flow rotor module. It specifically analyzes and designs system multi-structure, main function modules, the realization of key skills in the

completed modules and system test. At present, this system has practically applied in the PDL infrastructure maintenance base in Shanghai and Wuhan. And the practical application suggests, this system has reliable performance and has achieved the design targets, which won the praise of all users.

Key Words: passenger dedicated lines; construction management; Beehive; Workflow; JavaEE

第 1 章 绪论

1.1 课题研究背景及意义

近年来,随着大规模铁路建设,客运专线建设达到空前的规模,预计今年底,我国将建成总里程达 1.3 万公里客运专线。根据中长期铁路网规划《中长期铁路网规划(2012 年调整)》,到 2020 年,全国铁路营业里程达到 12 万公里以上,复线率和电化率分别达到 50%和 60%以上,主要繁忙干线实现客货分线,基本形成布局合理、结构清晰、功能完善、衔接顺畅的铁路网络。国家规划在路网总规模扩大的同时,突出客运专线的建设,建立省会城市及大中城市间的快速客运通道,规划“四纵四横”的客运专线以及经济发达和人口稠密地区城际客运系统。建设客运专线 1.6 万公里以上^[1]。

目前已经启动北京、武汉、上海、广州客运专线基础设施维修基地建设。根据客运专线列车高速度、高密度、安全稳定运行的特点,需要建设相配套客运专线基础设施维修管理信息系统。作为系统中关键模块之一的施工管理相对既有线要求更加精确化、复杂化。在当前客运专线大型工程项目施工管理中,施工组织方式采用传统的文字、表格等形式。这些方式不够直观,且影响各施工参与方的沟通与执行效果,降低了信息交流共享的有效性,进而降低施工管理的效率。建设新的管理信息系统符合我国客运专线现状和实际要求。客运专线的工程项目施工管理中,数据信息量大,关系错综复杂,同时各客运专线维修基地工程管理的计划流程中涉及的单位组织较多。采用工作流这一工具来实现数据的流转便于异地实体间的信息共享和交流,有助于任务分担和协同。

1.2 国内外研究现状

国外在工程项目管理信息化方面的研究和应用已经有 30 多年的历史。从施工总计划的编制到施工组织设计、全过程的投资、施工进度和质量控制都有完善的理论、方法和相应的软件产品。例如美国 Symantec 公司的 Timeline 软件, Welcome 公司的 Open plan 软件, Microsoft 公司的 Project 系列软件等^[2]。目前,国际上先进的项目管理软件正在朝着系统化、集成化、网络化和职能化的方向发展。在一些大型工程建设项目管理中得到充分应用

并获得巨大的效益。而国内在工程项目管理方面也根据我国国情研发了一些项目管理软件,取得了一定成就。如梦龙 PERT、同望 Project、广联达等。

workflow 方面的研究主要起源于 20 世纪 70 年代,早期的 workflow 源于办公自动化。在 1962 年, Petri 网开始作为流程描述的建模手段应用于 workflow。到 20 世纪 80 年代,商用 workflow 开始应用于图像处理和文档管理。到 1993 年, workflow 技术比较成熟, workflow 联盟成立。现在商用 workflow 国外主要有 WebWork workflow 管理系统,完全基于 web 技术实现的 workflow 系统。ORBwork workflow 系统,基于 CORBA 的分布式管理系统等^{[3] [4]}。国内的工作流系统在实现方面虽然起步较晚,但近几年发展迅速,国内已经有多家比较有代表的实现方案,比如上海东兰的 LiveFlow^[5],杭州信雅达的 SunFlow^[6]。神州数码 workflow EasyFlow^[7],上海协创软件的 JoinWork^[8]。

JBPM^[9]是一种基于 J2EE 的轻量级 workflow 管理系统。作为当前市场上扩展性最好的开源 workflow,其过程建模技术结合了 UML 活动图和状态图的技术,能实现和提高系统的可视化设计。所以采用 JBPM 从经济性和软件系统的功能性方面有着显著的优点。

1.3 论文主要研究工作

本论文主要实现了客运专线基础设施维修管理系统施工管理模块,施工管理子系统基于 Beehive 框架,由 7 个相对独立的模块组成,论文在研究了相关技术的基础上,针对系统建设的需要,结合 JBPM workflow 技术,重点分层介绍了施工组织设计的设计、实现以及计划上报审批流程的重点实现内容。本文主要研究工作如下:

介绍了 Java EE 架构, MVC 模式, workflow 技术与 workflow 管理系统,分析了 workflow 参考模型;研究了 Struts1.2 和 Beehive 框架, Jbpm workflow 引擎的体系结构、流程定义和流程调度。

分析了施工管理模块的需求,基于 Beehive 框架进行分层设计,将业务逻辑封装成基础服务层避免了在业务层进行大量 SQL 的编写。以施工组织设计编制和计划上报审批为例对分层设计进行详细说明。另外对基于角色的访问控制 RBAC 模型进行改进,设计符合我国铁路信息化建设需求的 RBAC 改进型模型---RUP 模型。以施工组织设计编制为例分层详细说明了模块的实现。针对运行揭示文档格式统一且数据量较大的特点,研究了 JXL 技术实现 Excel 格式文档的导入。以及使用 JBPM workflow 引擎实现施工计划的管理。最后完成对系统的测试。

1.4 论文结构安排

根据各项研究工作，本文共分为五章，各章具体内容如下：

第一章：主要阐述了本文的选题背景、技术的研究意义以及其国内外研究现状，并介绍了本论文完成的主要工作。

第二章：介绍系统主要使用的三个方面的技术，首先介绍了 Java EE 架构以及 MVC 框架，重点阐述了 Beehive 框架。然后介绍了 workflow 技术、workflow 管理系统，重点分析了 Jbpm workflow 引擎。

第三章：客运专线基础设施维修管理系统施工管理子系统的需求分析，分析了子系统需求与组成，对 7 个子模块的功能需求进行了分析。

第四章：客运专线基础设施维修管理系统施工管理子系统的设计，首先介绍了系统的网络架构，然后在 Beehive 框架的基础上进行分层框架设计，并以施工组织设计为例进行了各层的详细设计。

第五章：以施工组织设计为例，对施工管理子系统的实现进行详细的说明，系统中关键技术介绍 Excel 文件的导入、页面功能的权限控制、workflow 关键部分的实现，最后还完成了对施工管理子系统的测试。

结论和展望，对本文完成的工作进行了总结，并指出了未来研究的方向。

第 2 章 系统相关技术研究

2.1 面向 JavaEE 的轻量级应用框架 Beehive

2.1.1 JavaEE 技术介绍

JavaEE 是在 JavaSE 基础之上建立起来的一种标准开发架构，主要用于企业级应用程序的开发。在 JavaEE 的开发中是以 B/S 作为主要的开发模式，在 JavaEE 中提供了多种组件及各种服务^[10]，如图 2-1 所示。

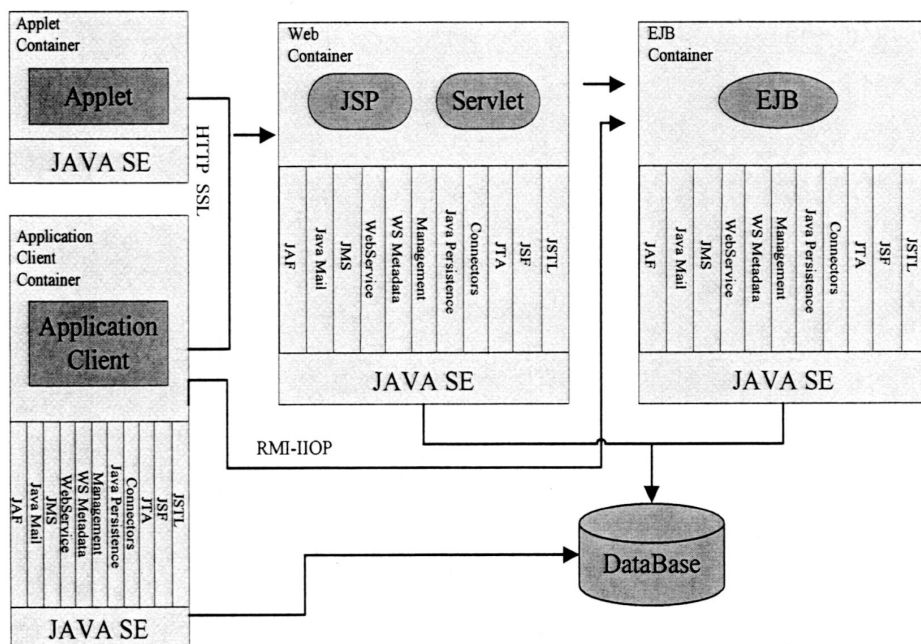


图 2-1 JavaEE 架构图

1、JavaEE 容器

容器主要负责组件的运行，JavaEE 一共提供了 4 种容器，即 Applet Container、Application Client Container、Web Container 和 EJB Container。各个容器负责处理各自的程序，互相没有任何影响，而如果要运行 Web 程序，则一定要有 Web 容器的支持。

2、JavaEE 组件

每一种 JavaEE 组件都表示一种程序的开发。如 Application 程序就是一种使用主方法运行的组件。在 JavaEE 中提供了 4 种容器，每一种容器中都运行各自不同的组件，在 Web 容器中运行的是 JSP\ Servlet 组件。EJB 组件

本身则提供的一个业务中心。

3、JavaEE 服务

JavaEE 应用广泛的原因在于其提供了各种服务,通过这些服务可以方便用户进行开发。例如如果要进行发送邮件操作,则可以使用 JavaMail 服务。在 JavaEE 中的主要服务有 HTTP、RMI-IIOP、Java IDL、JavaMail、JNDI、JAF 等。

2.1.2 基于 MVC 模式的 Beehive 框架

1、MVC 模式

MVC 设计模式最早由 Smaltalk 提出^[10]。它强制性地使应用程序的输入、处理和输出分开。MVC 设计模式被分为 3 个层,即模型层、显示层和控制层。它们各自处理自己的任务,各层的任务如下。

➤ 显示层(View)

主要负责接收 Servlet 传递的内容,并且调用 JavaBean,将内容显示给用户。

➤ 控制层(Controller)

主要负责所有的用户请求参数,判断请求参数是否合法,根据请求的类型调用 JavaBean 执行操作并将最终的处理结果交由显示层进行显示。

➤ 模型层(Model)

完成一个独立的业务操作组件,一般都是以 JavaBean 或者 EJB 的形式进行定义的。

在 MVC 设计模式中,所有的请求都要先交给 Servlet 处理,之后由 Servlet 调用 JavaBean,并将结果交给 JSP 中进行显示,如图 2-2 所示

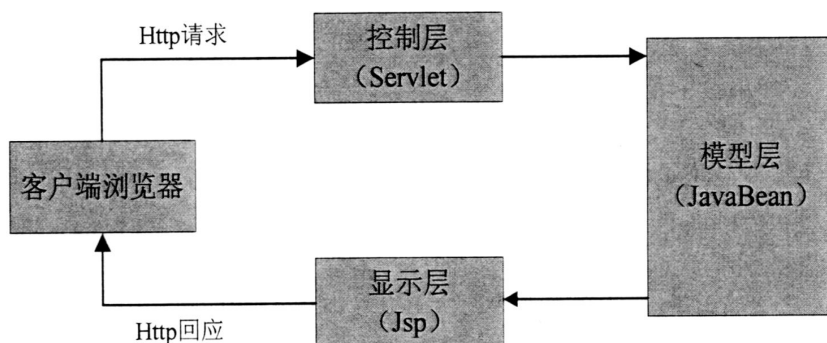


图 2-2 MVC 模式结构图

2、Struts 框架

Struts 是一个用来构建企业级 Java EE 应用程序的流行框架,通过使用

Struts, Java Web 应用程序的开发变得更加轻松且易于管理, Struts 是 MVC 设计模式的经典应用。在 Struts 中有自己的控制器,同时也提供了各种常用的页面标签库以减少 JSP 页面中的 Scriptlet 代码。Struts 属于在传统技术上发展起来的一种新的应用模式。Struts 中存在 Servlet, 需要由 struts-config.xml 进行控制, 而在 Struts 中的 Action 就相当于在基本 MVC 设计模式中一个个独立的 Servlet, 并且由 Action 调用模型层(JavaBean)完成一个个具体的业务功能。Struts 体系结构如图 2-3 所示:

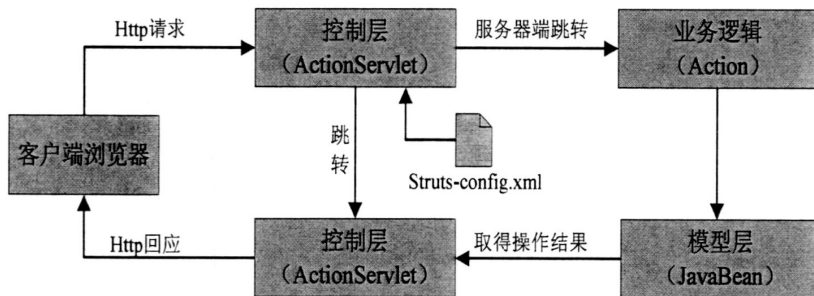


图 2-3 Struts 体系结构图

3、Beehive 框架

Beehive 是一个轻量级企业应用框架,用于简化 J2EE 以及 Web 服务编程。Beehive 框架分为三个子系统,分别是页面流、控件和 Web 服务^{[11][12]}。页面流是建立在 Struts 基础上的 Web 应用框架,它由各种注释驱动的控制类将应用程序的浏览逻辑、元数据和状态联系起来,它还提供了一系列 JSP 标签,并能与 JSF 和 Struts 集成。控件是一个轻量级的组件框架,将注释的使用和编程模型结合。降低了编程的难度。Web 服务在注释的基础上进行开发,降低了开发的复杂性。控件在应用中的地位是充当页面流和 Web 服务的提供者,即 MVC 中的 Model。Beehive 体系结构如图 2-4:

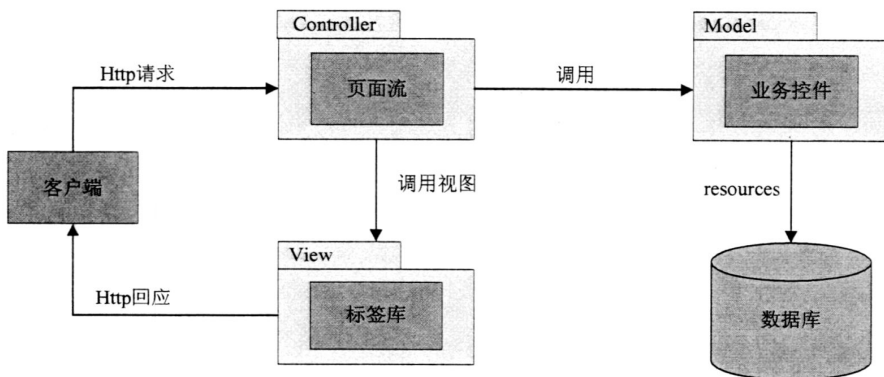


图 2-4 Beehive 体系结构图

2.2 workflow 技术

2.2.1 workflow 定义

按照国际 workflow 管理联盟的定义标准, workflow 是指计算机支持的业务流程的全部或部分的自动化^[13]。在流程执行过程中, 文档、信息资源或任务按照预定的规则, 通过网络为媒介流转, 实现组织成员间的协调工作, 以达到预期的目标。

workflow 管理系统是支持 workflow 实现的计算机软件系统, 是实现 workflow 定义、启动、调度、监控、分析等管理功能的工具集^[14]。

下面是与 workflow 相关的一些基本概念及其联系^[15]: 如图 2-5 所示。

(1) 流程定义: 业务流程的形式化描述, 用于支持系统建模和运行过程的自动化。流程可分解为一系列子流程和活动, 其定义主要包括描述流程起始、终止的活动关系网络, 以及一些关于个体行为的信息, 如组织成员、与 IT 相关的应用和数据等。

(2) 活动: 实现流程逻辑步骤的一项工作任务的描述, 一般分为手工操作和自动处理两类, 是过程执行中被 workflow 引擎处理的最小工作单元。

(3) workflow 参与者: 可执行流程活动实例的资源, 可分为人员、组织、角色和资源四类。

(4) 流程/活动实例: 实际运行中的一个流程或活动, 每个实例代表一个能独立控制执行、具有内部状态的线程。

(5) workflow 引擎^[16]: 为流程/活动实例的运行提供执行环境的软件服务, 提供按照流程定义来执行流程的功能。一个或多个 workflow 引擎构成了一个 workflow 域。

(6) 工作项: 被 workflow 参与者执行的活动实例的表示, 一个活动实例通常产生一个或几个工作项, 工作项通过赋予相关参与者的工作列表而被参与者感知。

(7) 工作列表: 流程参与者所拥有的用于感知流程工作项的列表, 是工作列表处理器(WorkList Handler)与 workflow 引擎的接口。

(8) workflow 控制数据: workflow 管理系统 Workflow Management System(WFMS)所拥有的, 不能被外部应用系统访问的, 用户控制 workflow 运行的服务的数据, 如系统所保存的各种有关流程实例状态数据。

(9) workflow 相关数据: WFMS 系统和外部应用系统所共有的, 用户控制流程实例迁移的数据, 该数据由 WFMS 和外部应用共同维护, 是两者的交互

数据。

(10) workflow 应用数据：外部应用系统自己的数据，它由各个外部应用所维护，不能被 WFMS 所感知，但 WFMS 可负责在不同的应用系统间传递应用数据。

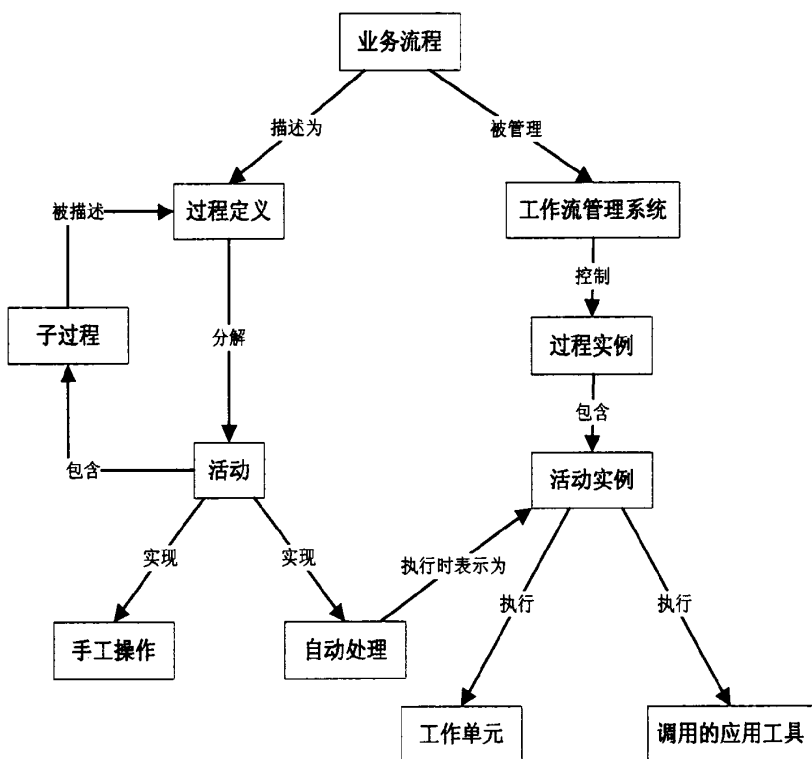


图 2-5 workflow 基本概念联系图

2.2.2 workflow 参考模型

为了更好地实现不同 workflow 系统之间的协作和信息交互， workflow 管理联盟定义了 workflow 参考模型^{[17][18]}，图 2-6 描述了它的基本部件和基本接口。

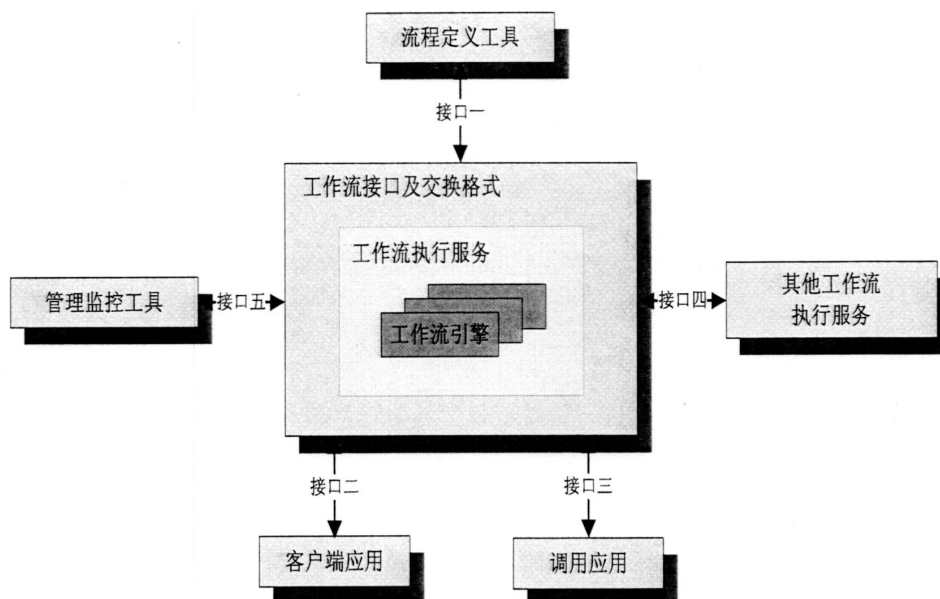


图 2-6 工作流参考模型图

该参考模型包括五个模型和五个接口。这五个基本模型是：

(1) 工作流流程定义：负责给出工作流程的定义，并以一定的数据格式提供给工作流引擎解释，执行。

(2) 工作流执行服务：工作流管理系统的核心，由一个或多个工作流引擎组成，负责创建、管理和执行工作流实例。各个工作流应用通过工作流应用程序接口(WAP)访问它。

(3) 供调用的应用：它是一些功能应用的集合，每一个应用可能负责工作流的某项任务，工作流引擎根据工作流的运行情况调用他们来完成工作流的执行。

(4) 工作流客户端应用：它可以通过工作列表访问接口访问工作列表库，而工作流引擎可以把任务分配到工作列表库，由用户去访问接受分配的任务。

(5) 工作流的管理和监控：负责管理监控工作流，包括用户管理、角色管理、运行记录、错误恢复、停止和删除工作流等。

2.2.3 JBPM 工作流技术

1、JBPM 工作流简介

目前业界的工作流产品实现主要遵循两个标准：BPEL 和 WFMC。BPEL 的全称是 Business Process Execution Language (业务流程执行语言)，IBM、ORACLE 等软件巨头都比较推崇 BPEL 标准。WfMC 全称为 Workflow

Management Coalition (工作流管理联盟), 采用该标准的工作流产品在国内是比较多的。JBPM 是一款开源的工作流产品, 它没有采用 BPEL 或 WFMC 标准去实现流程引擎, 采用的是一种轻量级的 XML 结构的流程描述语言 JPDL (JBPM Process Definition Language)。相比 BPEL 和 WFMC 两种标准而言, JPDL 语言更加简单, 也更容易读懂^[19]。

2、JBPM 流程定义

JBPM 是基于模型的组件, 流程实例都是依赖流程定义而执行的, 流程定义就是将业务流程形式化描述。JBPM 的流程定义采用了动态图的模型语义来描述^[20], JPDL 主要使用 7 种流转控制活动来实现流程的形式化描述^[21]。流转控制活动是组成一个业务流程定义最基本的活动, 利用这些活动可以构建任意的流程定义, 实现各种基本流程流转控制。JPDL 流程控制活动主要包括:

(1)start (开始活动): 指明流程实例开始的地方, 作为流程的入口。每个流程定义都只有一个 start 活动 (在 group 中的开始活动除外)。且 start 活动必须有一个流出转移。

(2)state (状态活动): 业务流程需要某些特定的操作后才能继续执行, 在这个操作之前流程处于等待状态。当流程运行到 state 活动时, 自动进入等待状态, 直到外来的信号触发。

(3)decision (判断活动): 根据条件在多个流程转移做出决定性的判断, 选择最先匹配的一个条件转移执行。

(4)fork-join (分支、聚合活动): 在需要并发执行流程时, fork 活动可以使流程出现并行的分支, join 活动可以使并行分支聚合成一个流程。

(5)end (结束活动): 当流程实例流转 to end 活动时, 实例会被终止。但到达 end 活动的实例中仍然活跃的流程活动 (如 fork-join 并发流转引起的流程分支) 将会被保留继续执行。JPDL 允许有多个 end 活动。

(6)task (人工任务活动): 用来处理需要人机交互的活动。可以分配任务给指定用户、实现任务分配处理器、任务泳道、任务提醒等功能

(7)sub-process (子流程活动): 允许在流程定义中调用其他的流程定义。流程的形式化描述就是有向图, 其本质是节点和有相向弧。JBPM 的节点 Node 就是流程控制活动, 它的相向弧就是节点的转移 Transition (转出或者转入), 使用节点和转移可以绘制任意一个业务流程。

3、JBPM 流程调度

流程调度就是流程实例从一个节点运行到另一个节点的工作机制^[22], JBPM 采用令牌 Token 传递的机制, JBPM 启动一个流程时, 首先创建一个

流程实例，并为这个实例创建一个根节点 Root Token，此时这个 Root Token 在 start 节点上，然后开始向下流转，如图所示，通过 Token 的 signal 方法，将 Token 放入到要转向的 Transition 对象中，再由 Transition 对象将 Token 交给下一个节点，这样就完成了流程调度^[23]。流程图如图 2-7 所示。

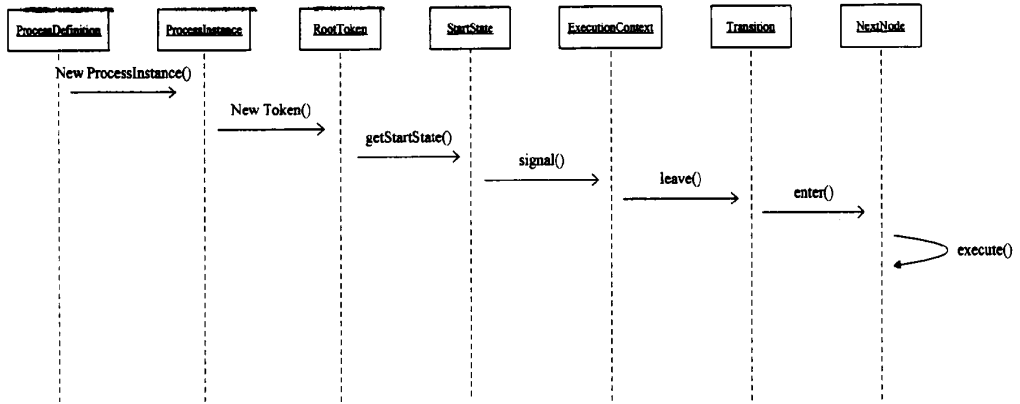


图 2-7 JBPM 流程调度时序图

第 3 章 施工管理子系统需求分析

3.1 总体需求

施工管理子系统是客运专线基础设施维修管理信息系统中综合生产管理子系统的子模块。该系统的总体目标是从提高客专基地大型养路机械施工过程精细化和规范化的角度出发,对计划编制、施工过程以及施工结果的精细化管理。从时间维度的年度、月份、施工日对施工计划进行层层拆分,从施工性质维度划分为大修清筛施工和维修施工分类进行管理。引入施工过程管理落实每日施工进度情况的监控,最后是施工验收管理对施工结果的把关。

3.1.1 客运专线维护体系与系统功能结构

铁道部在客运专线的组织机构设置方面进行了创新,分别设立了北京、上海、广州、武汉四大客运专线基础设施维修基地。按照客运专线固定设备属地化管理和专业化原则,客运专线固定设备维修管理实行铁道部、铁路局、段(下设车间、工区)三级管理模式。客运专线固定设备属地化管理由所在区域铁路局、段(工务段、电务段、供电段)负责,客运专线基础设施维修基地负责客运专线固定设备综合检测、大型养路机械作业等。铁道部基础设施检测中心负责代部分析综合检测数据,提供检测分析报告。各级部门主要负责如下,客运专线固定设备维修管理组织结构如图 3-1 所示。

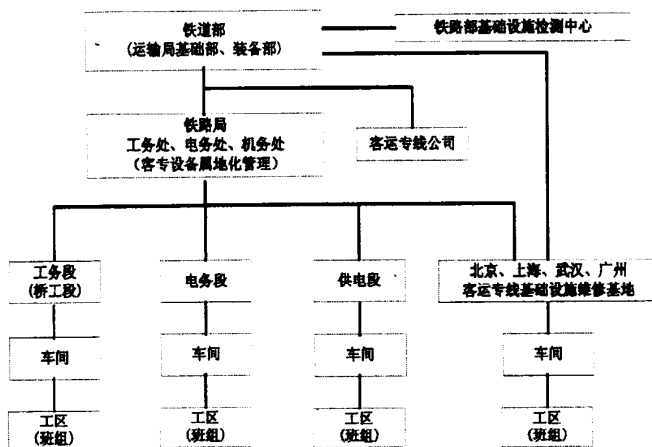


图 3-1 客运专线固定设备维修管理组织结构图

(1)铁道部:负责制定技术政策、技术标准、修程、修制,审定技术改造;

指导铁路局及客运专线基础设施维修基地业务工作；

(2)铁路局：负责所属客运专线固定设备维修和安全生产管理，指导所在地客运专线基础设施维修基地业务工作；

(3)段（下设车间、工区）：负责所属客运专线固定设备维修和安全生产管理；

(4)客运专线基础设施维修基地：负责覆盖区域内客运专线固定设备综合检测、大型养路机械运用和管理以及铁道部规定的工务、电务、供电设备维修工作；

(5)客运专线公司：负责客运专线资产管理，大修计划审批等；

(6)铁道部基础设施检测中心：负责代表铁道部分析各类综合检测数据，提供检测分析报告。

根据客运专线基础设施维修基地管理的实际需求，应用功能模块分为：“工务管理”、“电务管理”、“供电管理”、“综合生产管理”、“综合业务管理”五部分。综合生产管理主要负责生产和设备的数据管理、施工、安全、养路机械、监控，功能结构图如图 3-2 所示。

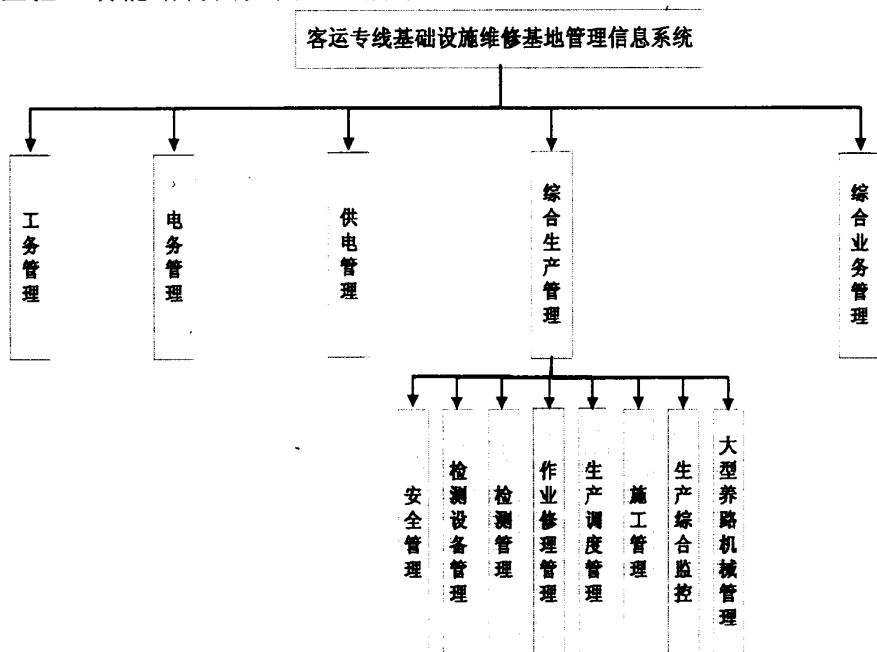


图 3-2 客运专线固定设备维修管理信息系统功能结构图

客运专线运行速度快，线路维护采用大型养路机械施工天窗作业方式组织施工，这类施工是一项系统工程，需要有关部门的协调配合，在施工中参与作业的大型养路机械多，实际作业前的准备工作量大、时间长、作业慢行距离长、作业程序多。并且此类施工对行车影响较大，每日施工完成后线路须限速运行，因此需要注重施工过程中的细节问题，把施工进度、质量的管

理和实际需求结合起来。

3.1.2 施工管理业务流程

在对施工管理精细化管理的实施过程中,首要的是客专维修基地根据铁道部和铁路局的年度计划进行各种施工计划的编制和拆分。具体流程如下:施工管理部门(客专维修基地)根据上级管理部门铁路局下达的施工计划编制大型养路机械施工方案,客专维修基地和铁路局相关处室需要参与各类计划流程审批和下达,对施工作业的过程进行调度管理和进度监控,监督施工单位对施工中出现的问題进行追踪和整改,最后对施工质量进行评定和验收。业务流程如图 3-3 所示:

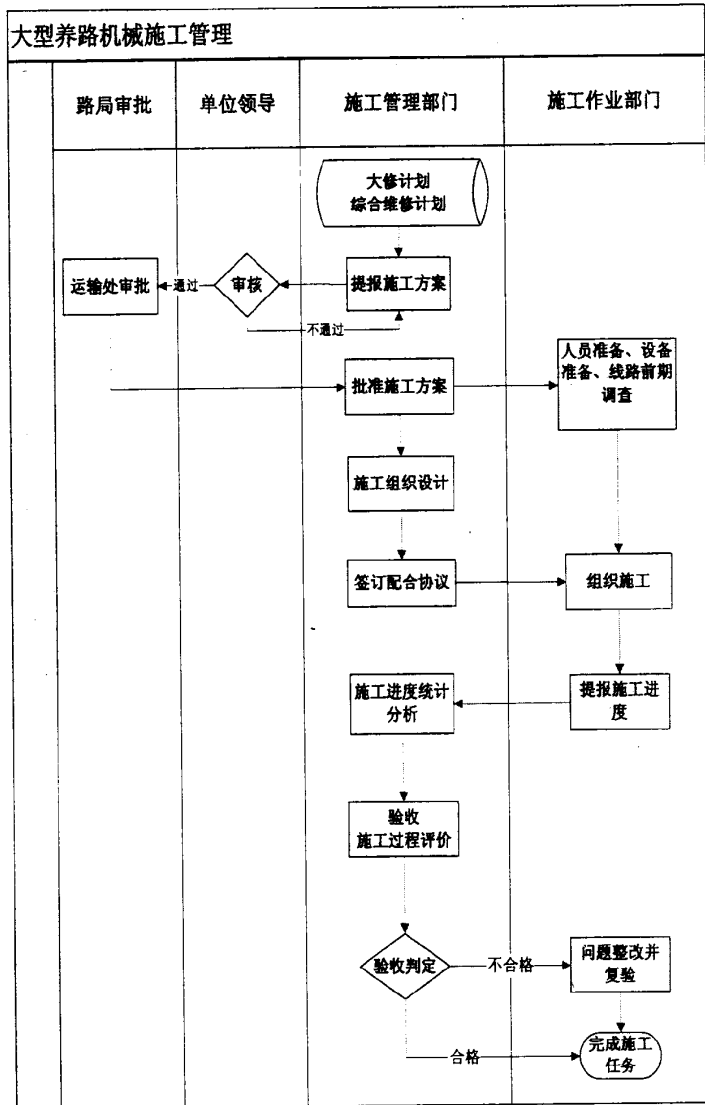


图 3-3 施工管理业务流程

根据对施工过程的需求调研确定施工项目主要包含以下功能点：建立施工项目台账、施工组织设计、施工计划管理、施工过程管理和施工验收和施工资料管理。

3.2 系统功能性需求

对功能性需求分析的第一步是用例分析。即对施工管理模块功能的描述。用例图的主要作用是描述参与者和用例之间的关系，有助于可视化的了解系统功能。事实上，用例将开发过程的各个部分串在一起。用例图可视化的表达了系统的需求，且把需求与设计分离开。

3.2.1 施工项目台账

建立施工项目台账是施工计划管理的第一个环节。施工管理部门客专基地根据铁路局下发的局级“年度轮廓施工计划”细化制定本基地级的“年度轮廓施工计划”，并发给各配合单位并跟各配合单位制定签订各项施工配合协议。维修基地是以工程项目为单位进行成本核算，在制定年度轮廓施工计划也就是阶段计划之前，首先要制定工程项目台账。工程项目台账就是某年度施工计划的工程名称、类别、总工作量，工作时等信息，它是该年度计划下属的月度计划、日计划的源头。如用例图 3-4 所示。

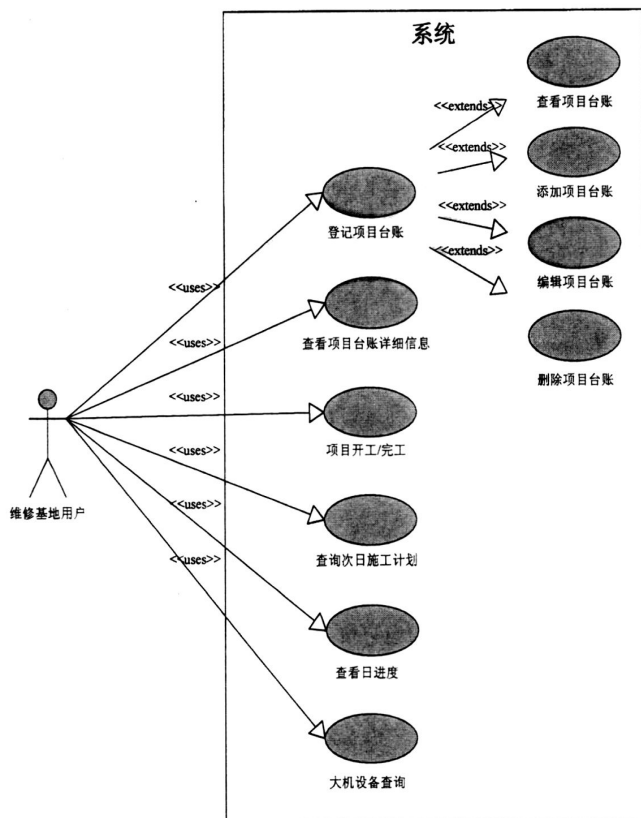


图 3-4 施工项目台账用例图

将工程项目台账拆分为施工阶段计划进而拆分为施工月计划、施工日计划来推进施工作业。施工完毕后，根据施工日情况逐级汇总出该工程项目的进度情况、投入成本情况以及设备使用情况等。施工项目台账的主要功能点如下：

(1) 查询、添加、编辑、删除项目台账：施工项目台账列表中，单击“添加”、“删除”、“修改”进行添加台账、删除台账、修改台账。填写查询条件，单击“查询”，显示满足条件的台账列表。查询条件有单位、工程项目名称、工号。

(2) 查看项目台账记录细节：施工项目台账列表中，单击“单位”查看项目详细信息。

(3) 查看日计划：单击“次日施工计划”查看次日施工计划信息。

(4) 查看日进度：单击“日进度”查看项目实际进展情况。

(5) 通过施工项目台账查看设备：查看项目使用设备情况。

(6) 设定开工/完工时间：设定开工和完工，以及具体开工完工的时间。开工完工时间只能设定一次，必须在完全确认情况下填写，先设定开工时间，开工时间确定后才能设定完工时间。

3.2.2 施工组织设计编制

在新线开始施工之前施工管理部门编制“施工组织方案设计”，从时间维度来看施工组织设计编制将年度轮廓计划细分为年度阶段计划。再从施工性质维度将年度阶段计划分为年度大修清筛阶段计划和年度维修阶段计划。

大修清筛主要进行线路道床石碴的清筛、捣固、道床的稳定及配碴、换枕与整形作业，对路基翻浆冒泥地段的整治。维修是以全面改善轨道弹性、调整轨道几何尺寸和更换、整修失效零部件为重点，以大型养路机械为主要作业手段，按周期、有计划地对线路进行的综合性维修，以及对全线进行有计划、有重点的经常性养护。施工组织设计编制的用例图如图 3-5 所示。

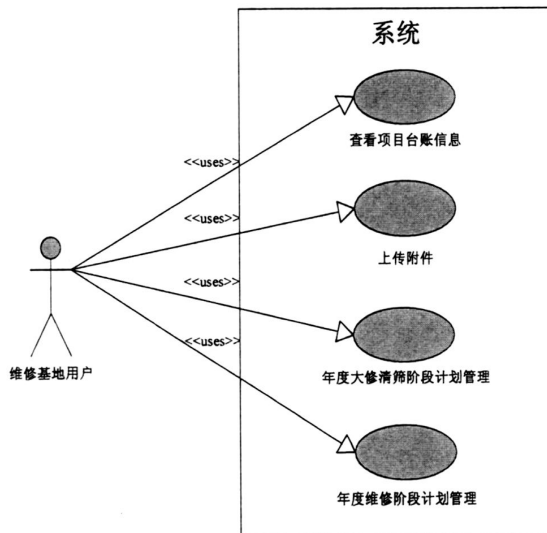


图 3-5 施工组织设计用例图

本系统确定施工组织设计包括的功能点有查询施工组织设计、附件上传、年度大修清筛阶段计划管理、年度维修阶段计划管理。具体活动场景如下：

(1) 查询施工组织设计

查询出符合条件的施工项目的信息，单击“查询”，显示满足条件的施工项目列表。查询条件为施工单位、工程项目名称、工号。

(2) 附件上传下载

上传和项目相关的施工文件。每个施工项目台账只能上传一个附件。

(3) 年度大修清筛阶段计划管理

单击“清筛”，进入相关联的年度大修清筛阶段计划管理信息列表。

(4) 年度维修阶段计划管理

单击“维修”，进入相关联的年度大修清筛阶段计划管理信息列表。

➤ 年度大修清筛阶段计划、年度维修阶段计划

对年度轮廓计划以及实际项目信息进行管理。包括的功能点有：查询年度轮廓计划、查看年度轮廓计划详细信息。用例图如图 3-6 所示具体活动场景如下：

(1) 查询年度轮廓计划

查询符合条件的年度轮廓计划信息。点击“查询”显示满足条件的年度轮廓信息。查询条件为单位、工程项目、阶段计划名称、线名、行别。还可以对计划信息进行编辑和修改、删除。

(2) 查看年度轮廓计划详细信息

单击“单位名称”可以查看年度轮廓计划的详细信息。

(3) 查看年度轮廓计划项目

点击“施工项目”进入施工推进计划项目列表查看施工项目信息，还可以对项目信息进行添加、编辑和删除。

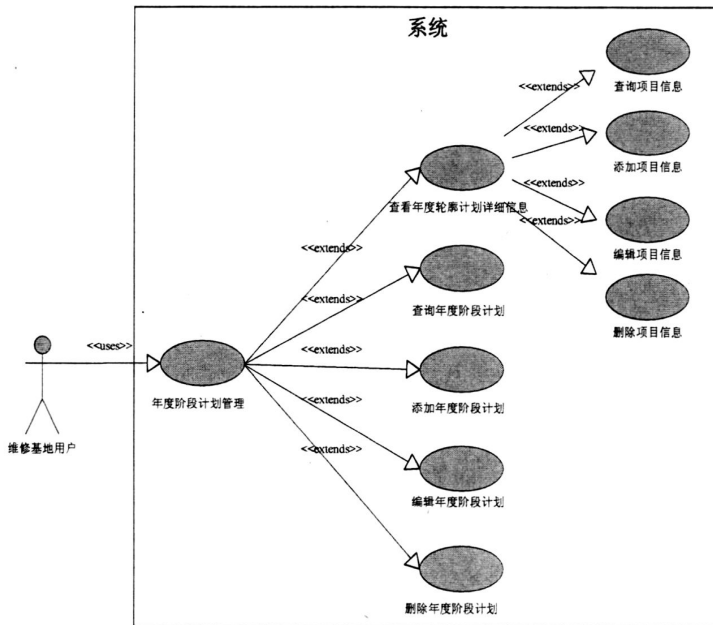


图 3-6 年度阶段计划用例图

施工组织设计将大修清筛年度施工阶段计划及其具体的推进计划施工项目、维修年度施工计划、附件文件等功能整合在一起。

3.2.3 施工计划管理

施工组织设计方案制定完毕后,根据阶段计划编制施工月度计划,在施工月计划申报审批流程中完成施工月度计划的要点,最后形成正式的施工月计划。根据施工月度计划,提前编制施工日计划,在经过施工日计划流程审批向调度要点施工,然后形成正式施工日计划。同时进行轨道车运行申请和路用车、宿营车等的转场申请以及卸车申请。

1、施工月计划

对施工阶段计划拆分后为施工月计划。施工月计划是一个需要多级领导审批的复杂流程,维修基地在编制施工月度计划通过层层审批后成为正式施工月度计划。业务流程图如图 3-7 所示,其业务步骤如下:

(1)客专维修基地根据正式施工年度计划编制施工月度计划,并上报业务主管部门工务处审批。

(2)业务主管部门工务处审批维修基地提交的施工月度计划,通过审批的施工月度计划提交给运输处。

(3)运输处审批工务处审批通过并提交的月度计划,通过审批的月度计划下达给工务处,并将正式月度计划交给调度所。

(4)工务处将正式施工月度计划下达给维修基地。

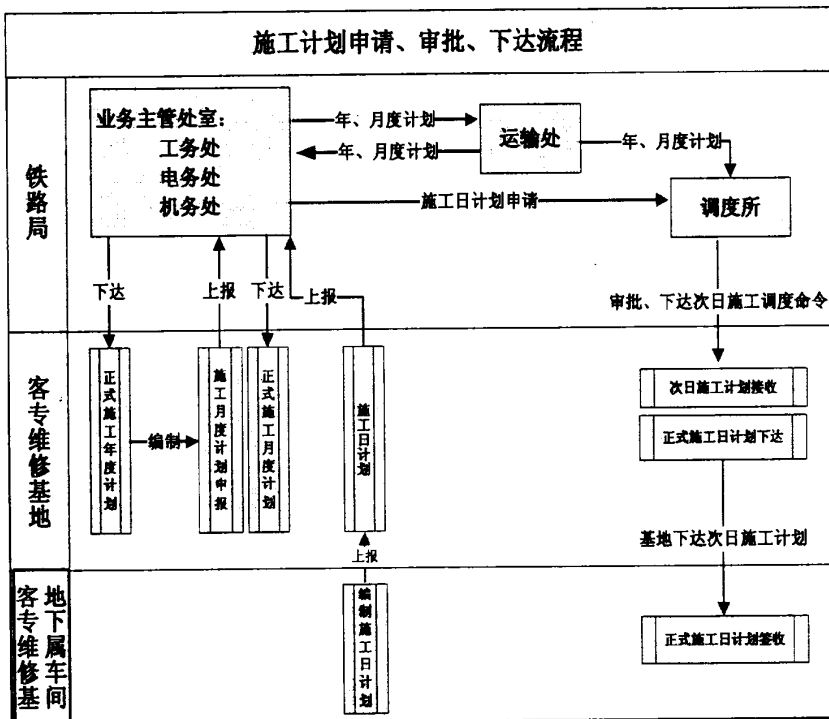


图 3-7 施工月计划、日计划流程

2、施工日计划

施工日计划由维修基地、基地下属车间、工务处、调度所等共同参与和完成。维修基地下属车间编制施工日计划通过各级审批后称为正式日计划，其业务步骤如下：

(1)施工作业部门车间编制施工日计划，并上报施工管理部门客专维修基地审批。

(2)施工管理部门客专维修基地审批施工作业单位车间提交的施工日计划，通过的审批提交给业务主管部门工务处。

(3)业务主管部门工务处审批维修基地审批通过的施工日计划。计划审批后提交给调度所。

(4)调度所审批工务处审批通过的施工日计划。下达正式施工日计划给维修基地。

(5)客专基地将审批通过的正式施工日计划下达给施工部门车间。业务流程如图 3-7 所示。

3、轨道车运行计划和车辆转场计划

施工作业部门（车间）在施工前三天提报计划，由施工管理部门（维修基地）联系铁路局驻台，将计划送报调度所；由行调台下调度命令；施工管理部门（维修基地）线路技术部调度员填写“轨道车运行表”，掌握转场动态，提供最新转场情况，并填写发车、到达时间。流程如图 3-8 所示。根据以上分析其业务步骤如下：

(1)施工作业部门车间填报计划，并上报给施工管理部门客专维修基地审批。

(2)施工管理部门客专维修基地审批施工单位车间提交的计划，将审批后的计划提交给调度所。

(3)调度所审批客专维修基地通过的计划，审批完成后提交给行调台，

(4)行调台审批调度所提交的计划，然后下达计划给维修基地。

(5)维修基地下达轨道车运行计划给下属施工车间。

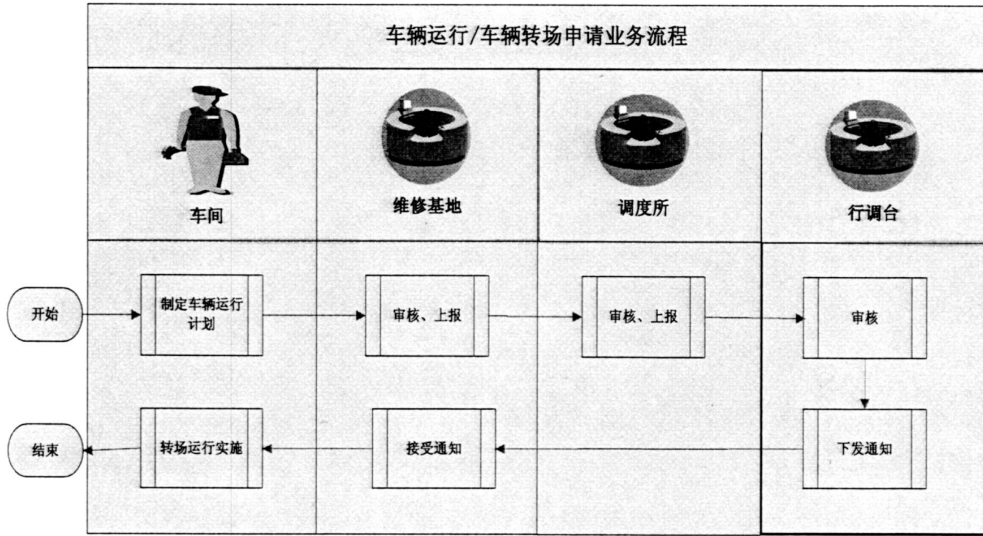


图 3-8 车辆运行、车辆转场申请业务流程

4、卸车计划

卸车计划的审批流程根据在施工过程中所卸材料的不同而划分为卸碴计划和卸枕计划。业务流程如图 3-9 所示。

➤ 卸碴计划上报审批流程：

- (1)施工作业部门（车间）于施工前将卸碴计划报施工管理部门。
- (2)施工管理部门上报工务处。
- (3)经工务处审核盖章后上报路局卸调台。
- (4)卸调台协调审批后上报行调台。
- (5)行调台下达行调命令给车间，然后车间实施卸碴计划。

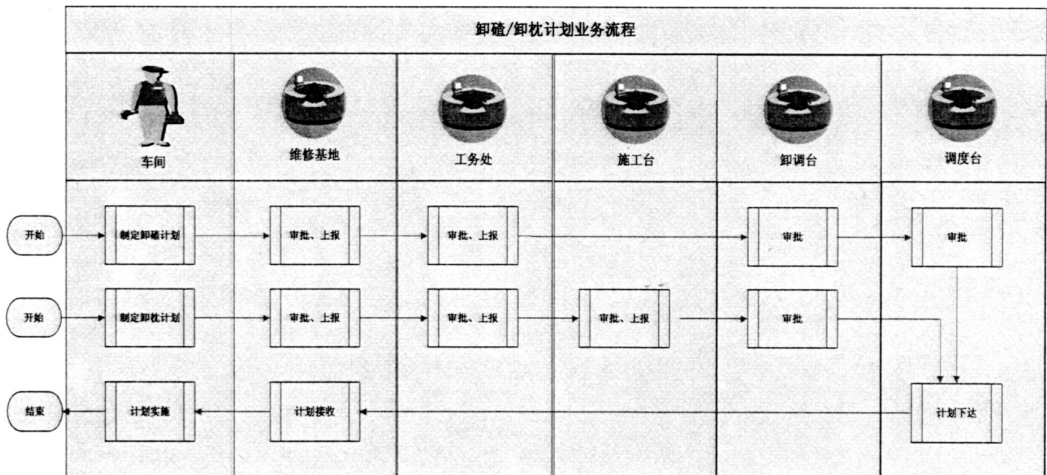


图 3-9 卸碴/卸枕计划业务流程

➤ 卸枕计划上报审批流程：

- (1)施工作业部门（车间）于施工前将卸枕计划报施工管理部门（维修基

地)

- (2)维修基地审批以后上报工务处，工务处审核盖章后报路局施工台。
- (3)路局施工台协调审批后上报路局卸调台
- (4)卸调台协调审批后再报送行调台
- (5)最后行调下达行调命令。施工车间在接到行调命令后实施卸枕计划。

3.2.4 施工进度管理

施工进度管理的目标是施工管理部门实时掌控施工进度状况，分析当前的施工组织是否与施工方案相匹配，根据分析结果及时调整施工生产总体安排。施工进度管理的主要功能点包括施工日进度、运行揭示、放散登记。

1、施工日进度

在施工过程中，各施工单位需要将实际施工情况填写施工日报并上报基地。客专维修基地对每日施工进度信息进行管理和掌控。施工日进度的功能点有查询施工日进度情况、查看施工日进度详细情况信息、复制施工日进度情况。具体活动场景如下，用例图如图 3-10 所示：

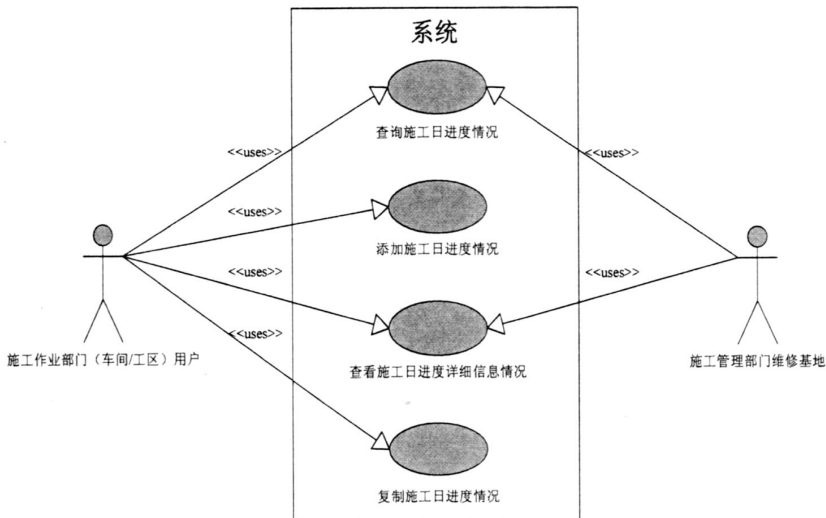


图 3-10 施工日进度用例图

(1)查询施工日进度情况

维修基地以及作业施工单位车间和工区用户查看施工日进度情况列表以及施工日进度详细信息。作业施工单位车间和工区用户还可以添加、修改和删除施工日进度情况。

(2)查看施工日进度详细情况信息

维修基地以及作业施工单位车间和工区用户查看施工日进度详细情况

信息。

(3)复制施工日进度情况

为方便新录入施工日进度，维修基地及施工单位用户可以复制施工日进度情况。

2、运行揭示

客专基地调度从路局调度处获取调度所施工台下发的施工运行揭示调度命令，据此编制基地营业线限速地段表，并上报路局工务处调度。运行揭示管理主要功能有查询运行揭示、导入运行揭示、查看运行揭示详细信息。具体活动场景如下：

(1)查询运行揭示

查询运行揭示是查询出符合条件的运行揭示信息。其中查询条件有单位、揭示时间。点击“查询”，显示满足条件的运行揭示列表。

(2)导入运行揭示

将运行揭示文件导入系统。运行揭示文件为 EXCEL 格式。

(3)查看运行揭示详细信息：

点击单位超链接，显示该单位对应的运行揭示详细信息。主要包括限速时间、限速里程、限速原因、限速速度、命令号码。

3、放散登记

放散登记的目的是管理无缝线路钢轨的温度情况，主要功能点包括查询放散登记列表、查看放散登记详细信息。具体活动场景如下：

(1)查询放散登记列表

查询放散登记列表是查询出符合条件的放散登记信息。其中查询条件有单位、放散日期、线名、行别、起始车站、终止车站。点击“查询”，显示满足条件的放散登记列表。

(2)添加放散信息

单击“增加”按钮，填写放散起始时期、放散终止日期、左股原锁定轨温，然后保存入数据库。

(3)查看放散登记详细信息

点击单位超链接，显示该单位对应的放散详细信息。

3.2.5 施工验收与竣工资料管理

1、施工验收

施工作业部门完成作业后，施工管理部门会同工务段、施工作业部门对施工作业进行验收并填写质量验收登记。施工管理部门在验收合格单位的合格证书上签认确认；施工作业部门对不合格的施工进行整改，整改后再会同工务段进行验收确认。

(1)查询验收信息

查询验收信息是查询出符合条件的验收信息。可以通过选择未验收和已验收条件来查看不同验收结果的信息。还可以设置的查询条件是：单位、施工日期、线名、行别、排序方式。

(2)验收施工

点击下拉框选择通过验收的项目，点击“验收”。验收情况显示为已通过验收。

2、施工检验

施工管理部门对计划验收和实际验收信息的管理。功能点包括查询检验信息列表、查询施工详细信息、查询施工检验详细信息。

(1)查询检验信息列表

查询检验信息是根据条件查询具体验收信息。可以设置的查询条件有：单位、工程项目、线名、行别。

(2)添加检验信息

单击“增加”按钮，填写计划起始车站、计划终止车站、实际起始车站、实际终止车站等信息，然后保存。

(3)编辑检验信息

点击“修改”进入编辑检验信息页面，修改计划起始车站、计划终止车站等信息，然后保存。

(4)查询施工检验详细信息

点击单位超链接，显示该单位对应的施工检验详细信息。

(5)施工日进度

点击“施工日进度”，显示该项目下的所有施工日进度信息。

3、施工资料管理

当某一施工阶段计划完成后，将竣工资料归档保存，以便以后查阅和统计分析以辅助决策。

(1) 查询施工项目列表

根据单位、工程项目名称、工号查询符合条件的施工项目列表。

(2) 施工文件上传与下载

点击“竣工文件”进入竣工文件管理页面，可以上传、下载、修改、删除文件操作。

3.2.6 计划申报审批 workflow

计划申报审批管理是对各类计划的维护和归类。负责各类计划的流转。根据计划审批与登录用户的关系，计划申报审批管理中的计划分为我的计划、待审计划、已审计划三部分，用例图如图 3-11，具体功能如下：

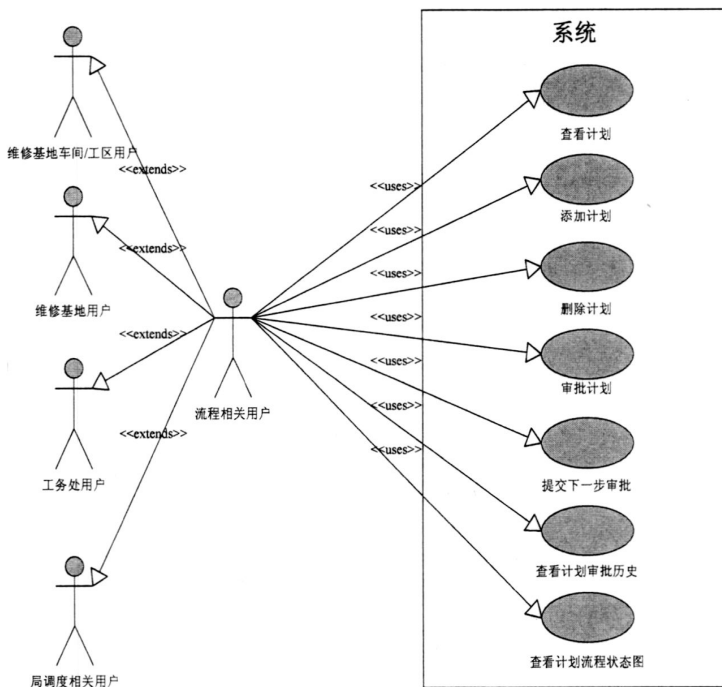


图 3-11 计划申报审批用例图

(1) 计划列表

显示登录用户所创建的计划列表，并显示计划的审核状态。用户可以创建各类计划并将不同类型的计划提交给不同的用户。

(2) 待审计划列表

显示登录用户需要审批的计划列表。用户填写审批意见以后可以将计划回退或提交给下一个用户。

(3) 已审计划列表

显示登录用户审批过的计划列表并可以查看计划的审批历史和当前状

态。

各类计划的业务流程都是按照客专维修基地施工作业的规范和制度制定，具有严格先后顺序和各自部门的责任分工。传统的工作流流转方式通过编程的方式设计业务流程。这种方式在业务流程发生变化或者需要新的业务流程时必须修改程序。显然这种方式不能满足系统多种计划流程的需求。因此，系统实现可定制的流程管理功能，使用户可以方便的定义和调整业务流程。

本系统采用 workflow 技术实现各类计划的申报审批功能。workflow 管理功能由系统管理员完成。主要负责流程的定义、部署、查看、删除。针对不同的流程定义不同的表单。由此可知计划申报审批 workflow 的功能如下，其用例图如图 3-12 所示：

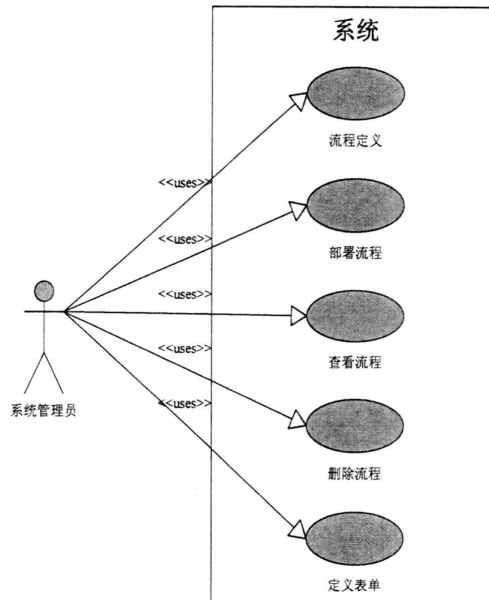


图 3-12 计划申报审批 workflow 管理用例图

- (1)流程定义：根据施工过程中各类计划的业务流程定义 workflow；
- (2)流程部署：将已经定义好的 workflow 部署到系统中；
- (3)查看流程：查看系统中已经部署的流程定义和流程图图片；
- (4)删除流程：删除系统中已部署流程定义的相关信息。

3.3 非功能性需求

施工管理模块是客运专线基础设施维修基地维修管理系统的一个子系统，考虑到系统的可靠性、稳定性、安全性，系统还需要达到以下几个目标：

- (1)标准化

系统各模块之间及与外部系统之间信息交换采用统一的标准或协议。

(2)可扩展性、可伸缩性、可移植性

采用模块化设计结构，在考虑满足当前各种业务实际要求的同时，充分考虑到将来用户数量增加、业务种类和业务量增长的需要。另外设计时，需要考虑到硬件配置和环境发生变化的可能性，最大限度的提高软件的可移植性。采用数据驱动的设计方法，通过调整系统参数，实现系统功能的变更。

(3)安全性

提供设备、应用、用户数据等方面的安全保障。

(4)可实现性、经济性

充分考虑和利用各种现有资源，设计时，既要考虑到系统的先进性和有效性，又要考虑其实现的可能性和经济性。

第 4 章 施工管理子系统设计

4.1 网络架构

客运专线基础设施维修基地利用客运专线数据网接入铁路局网络汇接点，并在铁路局汇接点实现与工务管理信息系统等既有网络互联互通，满足业务应用需求；客运专线基础设施维修基地下属的异地车间、工区（班组）与维修基地联网。

4.1.1 广域网

基础设施维修基地通过广域网上联接入所在铁路局网络，并连接其异地下属的维修车间和工区，维修基地同地下属维修车间和工区采用局域网与维修基地连接。连接图如图 4-1 所示：

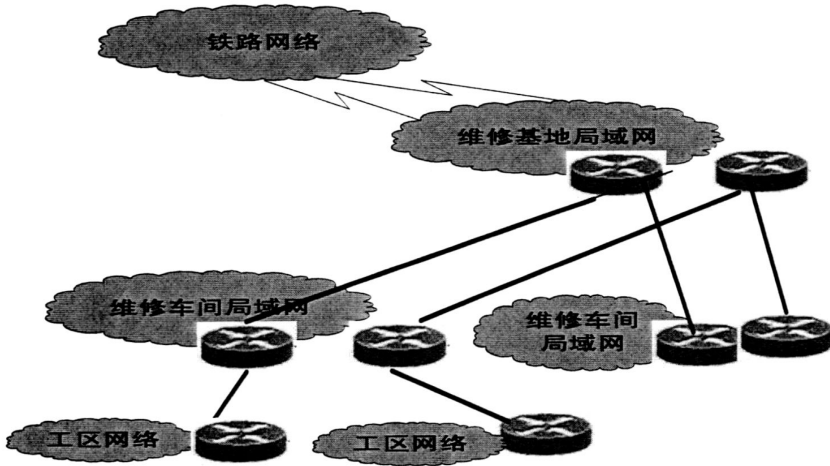


图 4-1 客专基地局域网与铁路广域网联接示意图

4.1.2 客专维修基地局域网

基础设施维修基地局域网通过广域网上连铁路局，内部划分为服务网和生产网，不同网络区域设置防火墙，配置安全策略，控制区域之间的相互访问。内部服务网以核心交换机为中心，连接各楼层或各部门的汇接交换机，并统一规划 IP 地址。

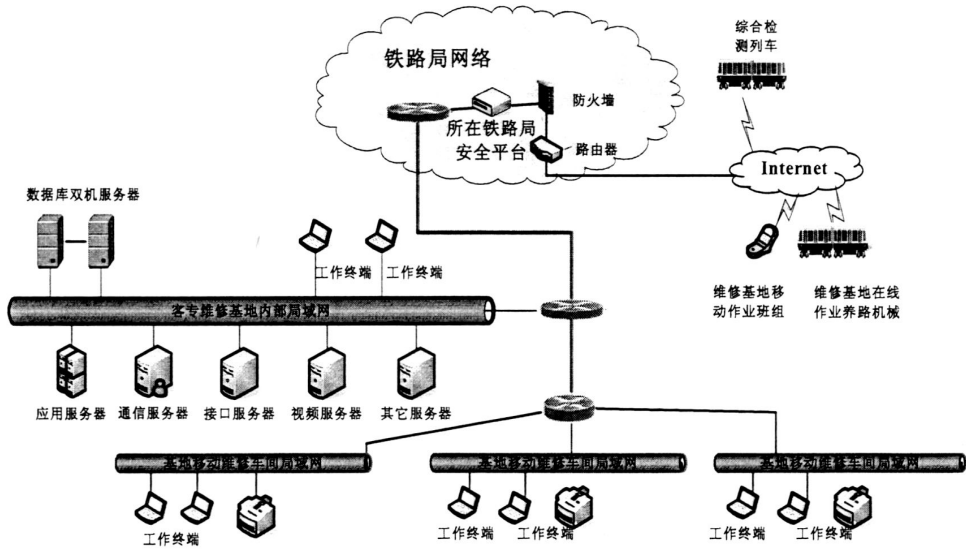


图 4-2 客专维修基地局域网示意图

4.1.3 维修基地下属车间（工区）局域网

维修车间（工区）局域网通过广域网连接到维修基地，下联内部服务网核心交换机。核心交换机连接各个分散用户。各信息系统数据汇聚到车间核心交换机和防火墙，统一利用广域网通道传输数据。

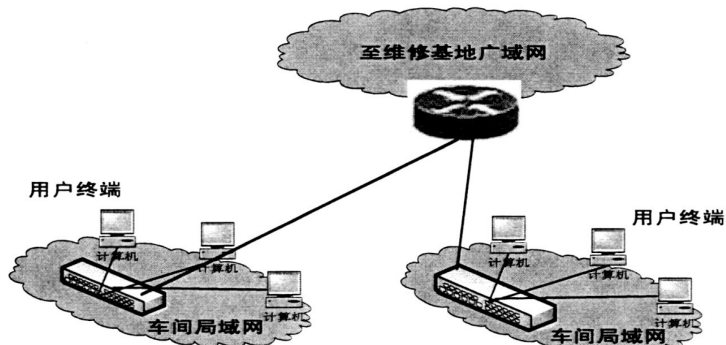


图 4-3 客专维修基地下属车间（工区）局域网

4.2 Web 开发框架设计

框架是解决相同或相似类型问题的方案。框架的特点是不针对客户，而是提供给开发者使用的基础性的服务，与具体行业没有绑定，具有一定的规范。在软件设计中，良好的框架设计有利于确定系统整体结构，层次划分，不同部分之间的协作，也有助于提高开发速度以及提高系统的可维护性。框

架比架构更具体，更偏重于技术。比如 MVC 模式可以通过多种框架来体现 [24]。

4.2.1 系统的可复用性与“开-闭”原则(OCP)

对于面向对象的软件系统设计而言，在支持可维护性(Maintainability)的同时，提高系统的可复用性(Reuseability)是一个核心问题，一个易于维护的系统通常就是复用率较高的系统；而一个复用性较好的系统，就是一个易于维护的系统 [25]。软件复用的优点：(1)提高软件生产率，减少开发时间；(2)提高软件质量，开发出来的软件可靠性高；(3)降低开发风险；(4)简化软件开发流程，使得软件开发易于管理；(5)降低维护难度，工作量和费用，提高了软件系统效益；(6)促进软件开发过程的标准化；(7)易于提供文档资料。

“开-闭”原则 [26] 是指：一个软件实体应当对扩展开放，对修改关闭。也就是说在设计一个模块的时候，应当使这个模块可以在不被修改的前提下被扩展。换言之，可以在不必修改源代码的情况下改变这个模块的行为。软件系统的需求随时间的推移而发生变化。在面对新的需求时，系统的设计必须是稳定的。满足“开-闭”原则设计可以给软件系统两个优点：(1)通过扩展已有的软件系统，从而提供新的行为，以满足对软件的新需求，使变化中的软件系统有一定的适应性和灵活性；(2)已有的软件模块，特别是最重要的抽象层模块不能再修改，这就使变化中的软件系统有一定的稳定性和延续性 [25]。

4.2.2 基于 Beehive 的应用开发框架设计

作为一个轻量级的应用程序框架，Beehive 为 Web 应用开发提供了大量实用的服务，如为 JDBC、JMS 队列提供了良好的封装，为业务逻辑提供安全性事务访问，为表示层提供 MVC 框架。仅用 Beehive 框架，开发人员就可以构建完整的企业级 Web 应用。按照 Web 开发多层开发原则，把基于 Beehive 框架 Web 应用分为表示层、业务层、数据库访问层、数据库层，如图 4-4 所示：

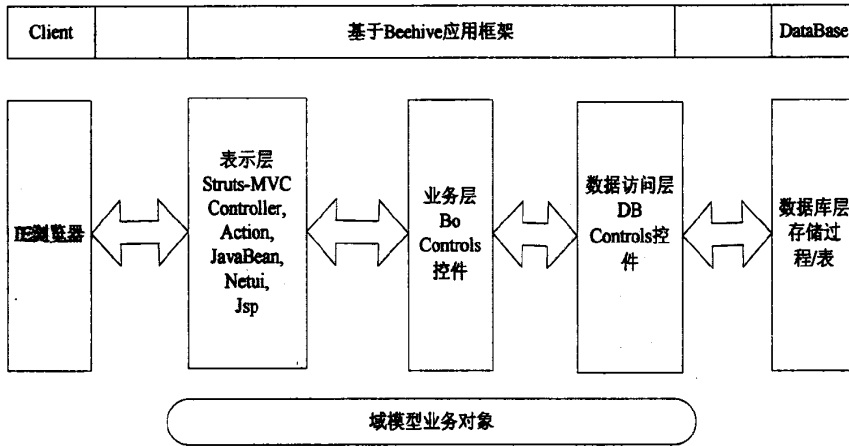


图 4-4 基于 Beehive 分层框架设计图

由上图可以看出：表示层提供用户与系统交互的界面，作用是输入和返回结果。业务层实现功能模块的各业务逻辑。数据库层主要实现绑定数据源、建立数据库连接、发送请求和返回结果集等。

该框架的优点在于采用多层分布式架构，分层体系降低了层与层之间的耦合度^[27]，Controls 核心框架支持一个控件接口与其一个或多个实现之间的松散耦合，也可以在运行时重写应用于控件的元数据。在开发中可以通过这个可插入的元数据机制，在运行时重新设置数据库控件的 JNDI 数据源名称而不必修改服务中的业务逻辑。对服务进行修改而不影响客户端程序很好的满足了“开-闭”原则^{[28][29]}。页面流技术使页面和应用逻辑分离，开发人员无需考虑繁琐的配置文件的设置，专注于系统的业务逻辑设计^{[30][31]}。

框架的不足在于：由于客运专线基础设施信息系统业务比较复杂，开发人员在业务层需要编写的 SQL 语句较多。而大量对数据库的操作除了少数情况是拼接 SQL 对单个数据表进行增、删、改、查外，大多数是通过传入参数拼接 SQL 语句调用数据库存储过程。因此，有必要在业务层业务逻辑的可重用性方面进一步深入改进设计。

4.2.3 基于 Beehive 的应用开发框架改进设计

基于提高业务层业务逻辑可重用性的目标，单独设计了业务基础层用于封装大部分业务逻辑，主要是对增、删、改、查、批量增加、批量修改、批量删除的封装。根据读写操作的不同设计 WriteControl 和 ReadControl 数据库控件。ReadControl 主要负责读取记录、读取记录的数量、为插入动作准备序列值。WriteControl 主要负责增加、删除、修改操作。

这样在实际开发中，业务层 BO 控件可以直接引入业务基础层控件，而

直接传入存储过程的参数就可以通过调用数据库控件进而完成对数据库表的操作。因此开发人员在业务逻辑层几乎不用编写 SQL 语句，对于开发效率的提高效果显著，改进框架图如图 4-5 所示。

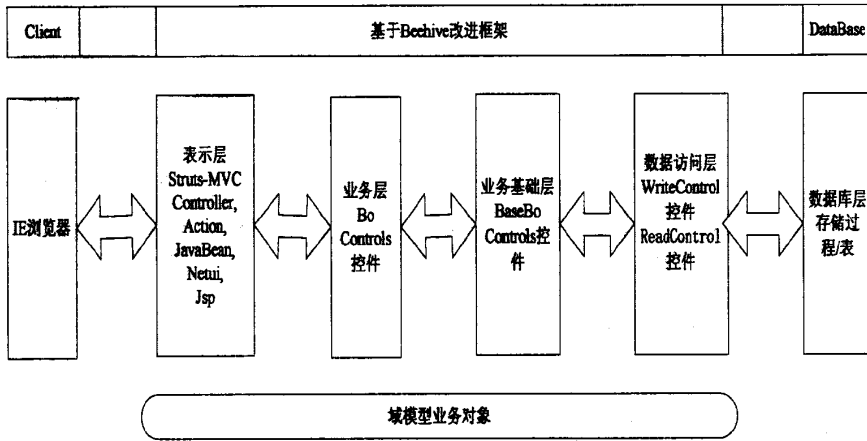


图 4-5 基于 Beehive 分层设计框架改进图

4.3 系统功能模块设计

经过以上对施工管理总体功能的分析，将该系统划分为施工项目台账、施工组织设计、施工计划管理、施工过程管理、施工验收管理、竣工资料管理、计划上报审批 7 个功能模块：在施工管理系统中，施工组织设计和施工计划上报审批流程是其核心部分。本文的模块设计就以这两者为例进行系统的设计描述。由于每个功能模块的框架和软件设计步骤都是类似，其他模块的设计可以按照本设计方式完成。如图 4-6 所示。

4.3.1 施工组织设计编制详细设计

施工组织设计主要包含 4 个菜单项，是系统大修清筛年度计划、维修年度计划、大修清筛项目推进计划、施工组织文件数据的录入接口。主要由客专维修基地用户操作。用来完成施工组织具体设计工作。下面介绍各菜单的设计说明，其中以查询施工组织设计列表菜单为例结合 4.3.3 节介绍的 Web 开发框架进行详细设计说明：

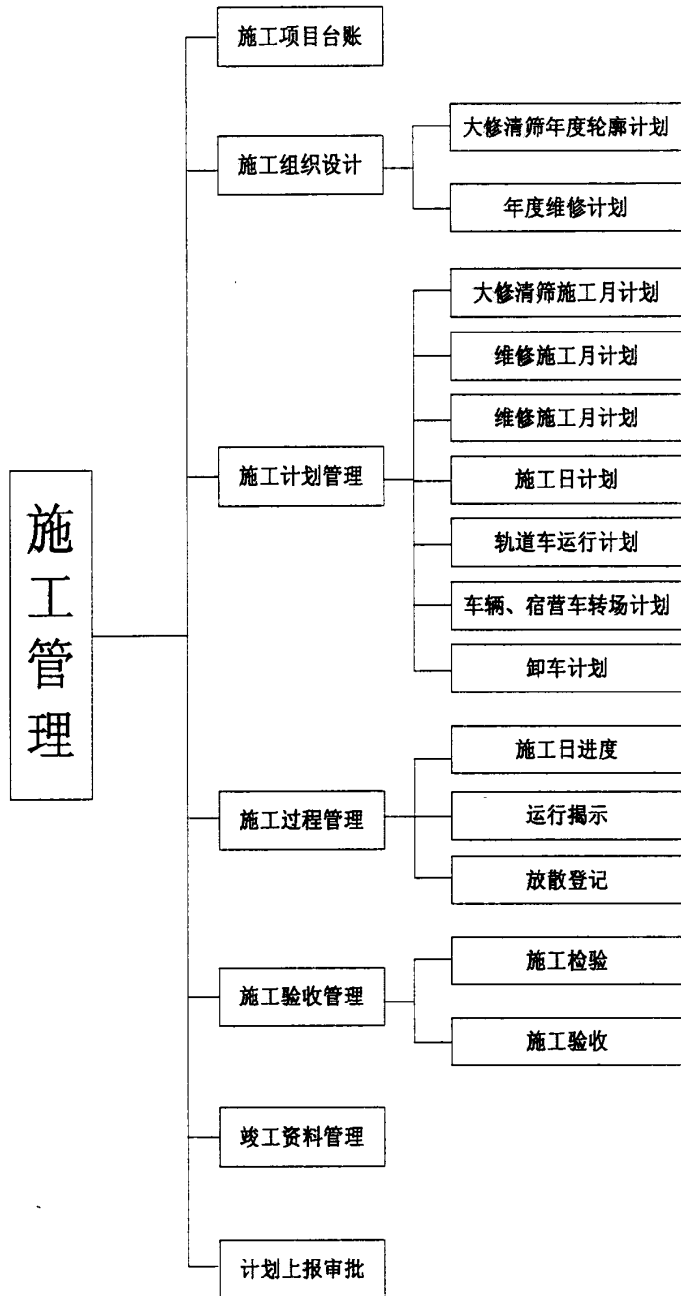


图 4-6 施工管理功能模块图

1. 查询施工组织设计列表：用于查询施工项目台账数据，从施工项目表中读取客专维修基地用户在施工项目台账编制中提交的施工项目台账记录以列表分页的形式显示在页面上，列表显示的信息包括单位、施工类别、项目名称、工号、总工作量、总工期、总工时、预算总额、开工时间、完工时间。在查询施工项目台账的时候，可以通过单位、工程项目名称、工号中的一项或多项进行关键字查询，检索结果以相同列表分页显示。点击单位链接，可以查看施工项目台账的详细信息。点击“附件”，可以对施工组织设计用到的文件进行上传，方便存档。

2. 大修清筛年度轮廓计划管理:用于查询、修改、删除大修清筛年度轮廓计划数据,并提供施工推进计划项目的链接。从大修清筛年度轮廓计划表中读取客专维修基地用户提交的清筛年度轮廓计划记录以分页的形式显示在页面上,列表显示的信息包括单位名称、工程项目名称、计划名称、起始日期、终止日期、线名、行别、起始车站、终止车站。可以通过单位、工程项目、阶段计划名称、线名、行别中的一项或多项进行关键字查询,结果以相同列表分页显示。

3. 大修清筛年度轮廓计划施工推进项目计划管理:用于显示当前大修清筛年度轮廓计划的详细信息和查询、修改、删除施工推进项目计划数据。从施工推进项目计划表中读取客专维修基地用户提交的施工推进项目计划记录以列表形式显示在年度轮廓计划的下方,列表显示的信息包括阶段计划内容、计划工作量、计划日进度、计划点数、备注。

4. 维修年度轮廓计划:用于查询、修改、删除维修年度轮廓计划数据。从维修年度轮廓计划表中读取客专维修基地用户提交的维修年度轮廓计划记录以分页的形式显示在页面上,列表显示的信息包括所属单位、施工台账、计划名称、线名、行别、维修项目、探伤遍数、延展长度、道岔总数。可以通过操作单位、施工台账、维修期限、线名、行别、起始车站、终止车站、计划名称、维修项目、施工年度中的一项或多项进行关键字查询,结果以相同列表分页显示。点击单位链接,可以查看维修年度轮廓计划的详细信息。

在软件工程设计中,对于描述对象之间的交互时,常以时序图来描述对象与对象之间的信息。作为交互图的一种^{[32] [33]},序列交互图按照时间顺序从上往下显示使用案例。查询施工组织设计列表的序列图如图 4-7 所示:

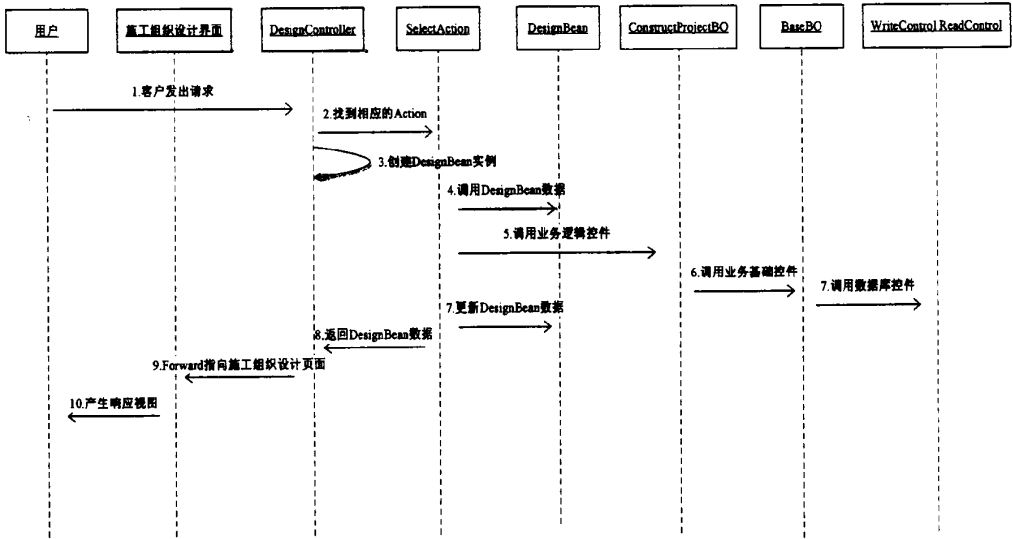


图 4-7 施工组织设计编制时序图

➤ 表示层设计

施工组织设计操作请求皆来自页面的请求，所以施工组织设计操作功能设计在表示层。在表示层设计了一个 DesignController 来充当控制器，完成页面用户对施工组织设计的动作请求。DesignController 是从 PageFlowController 扩展而来，PageFlowController 提供了有用的基类功能，如支持登录和注销。在 DesignController 中设置了多个执行动作 Action 及对应的页面跳转，Action 主要有：获取施工台账列表、根据查询条件得到施工台账列表、转到上传下载页面前动作、上传文件、下载文件、求该单位的顶级直属单位。每个 Action 都是一个 Java 方法。Action 均使用一个 FormBean 即 DesignBean 对客户端的表单数据进行封装。同时 DesignController 需要导入 ConstructProjectBO 等业务控件来对业务逻辑进行调用。

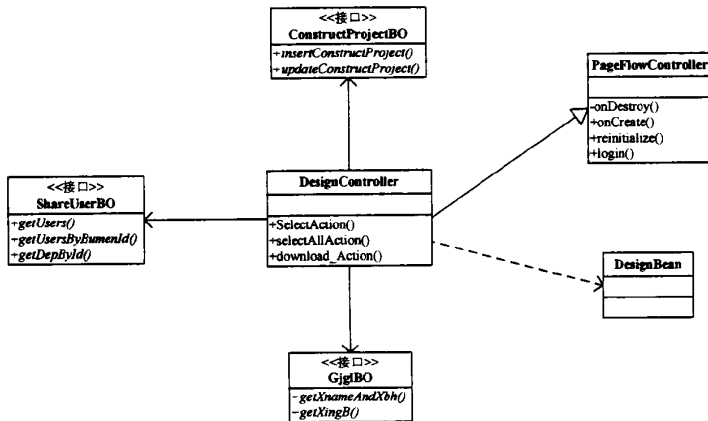


图 4-8 施工组织设计表示层的类图

表示层的时序图^{[34][35]}如图 4-9 所示:

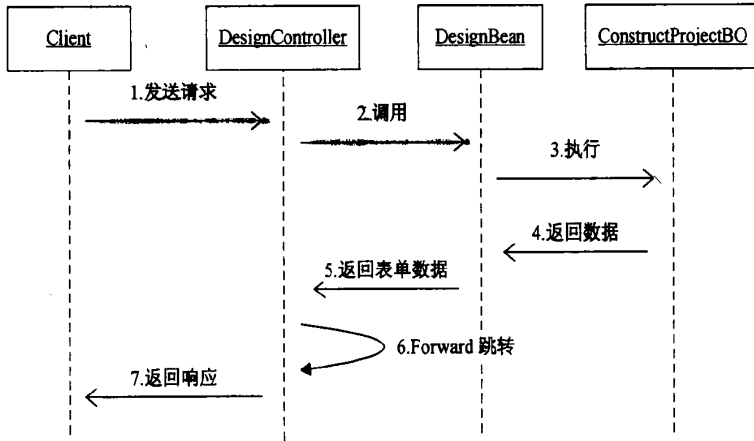


图 4-9 施工组织设计表示层的时序图

► 业务层设计

有关查询施工组织设计列表的业务逻辑都设计在 ConstructProjectBO 这个控件接口中，在获取单位的父节点数据和根据单位 ID 获取单位数据时需要分别调用 GjglBO 控件和 ShareUserBO 控件。业务逻辑的实现在 ConstructProjectBOImpl 类中。在查询施工组织设计列表时，我们需要按照单位、工程项目名称、工号关键字查询工程项目台账或者直接查询所有工程项目台账信息，所以在接口中设计 getConstructProjectBy()方法。传入该方法的参数为项目台账表对应类和分页信息，根据对应类中设置不同的值从而取得不同的查询结果。

在查询施工组织设计 getConstructProject 域模型中，系统对其业务逻辑实现的类图如图 4-10 所示。

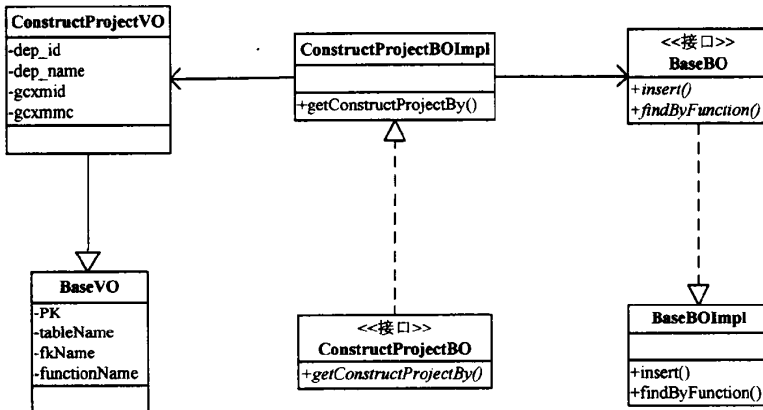


图 4-10 施工组织设计业务层的类图

如上图所示：查询施工组织设计列表业务逻辑中设计了 ConstructProjectBO 接口, ConstructProjectBOImpl 实现了 ConstructProjectBO 接口。BaseBO 控件中封装了对数据库的操作方法，传入 BaseBO 的参数 ConstrucProjectVO 必须要实现 BaseVO 类，并在 ConstructProjectVO 中给 BaseVO 中的基本属性值赋值，属性值为 ConstructProjectVO 所对应的表名、主键、外键名、查询单条数据存储过程名字、查询多条数据存储过程名字。对数据库操作方法的具体实现在 BaseBOImpl 类中。查询方法是根据传入的表名和参数利用函数 SQLHelp() 拼接好 SQL 语句传入 findByFunction 等方法，然后再调用 ReadControl 进行查询获得结果集。添加、修改、删除多个值的方法都调用同一个方法 ExecuteArrayUpdate() 执行，BaseBOImpl 类中插入方法的活动图如图 4-11 所示：

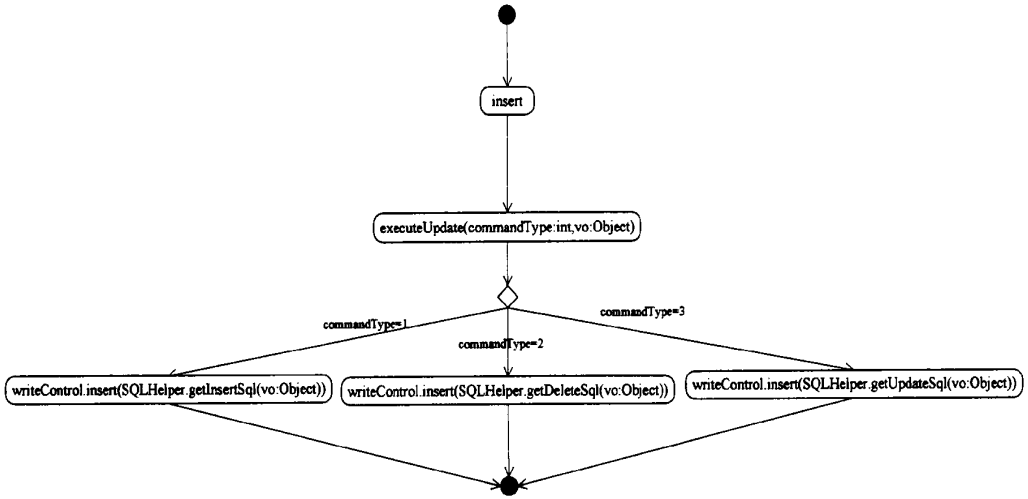


图 4-11 BaseBOImpl 中插入方法的活动图

业务逻辑层的时序图如图 4-12 所示：

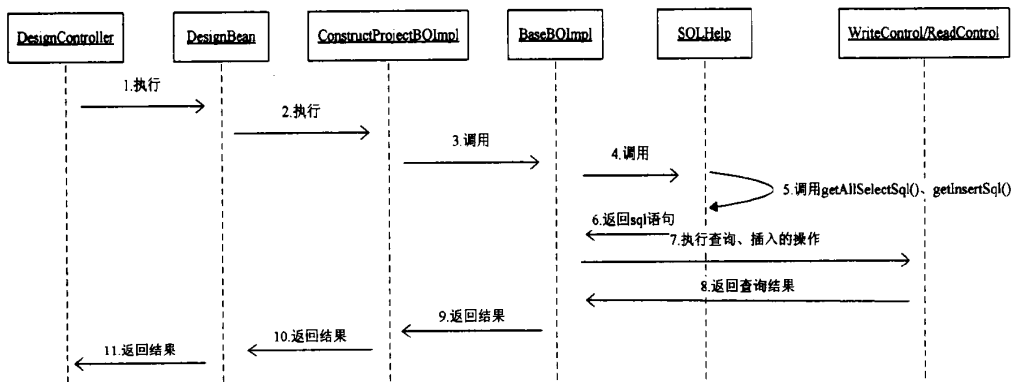


图 4-12 施工组织设计业务逻辑层时序图

➤ 数据访问层设计

数据层采用 WriteControl 控件和 ReadControl 控件实现。这两个控件都通过调用 Beehive 的系统控件 JdbcControl 来执行对数据库的操作。由于 JdbcControl 与其实现之间是动态绑定的，在 JdbcControl 控件可以方便修改 JNDI 数据源名称而不用设置配置文件。数据访问层类图如图 4-13 所示：

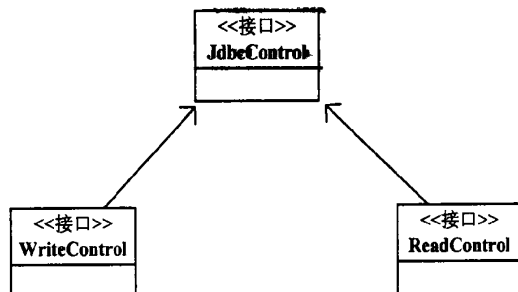


图 4-13 施工组织设计数据访问层类图

4.3.2 施工日计划上报审批流程设计

从第三章对施工计划上报审批的需求分析可以看出，施工管理模块采用 workflow 技术完成施工计划的业务流转和控制，由于上报审批的施工类计划比较多，而施工日计划是各类计划中的核心作业计划，因此以施工日计划详细设计为例作为描述说明。

1. 施工日计划工作流程设计

施工日计划工作流程使用开源 workflow 引擎 JBPM 进行日计划的业务流转和控制，流程定义语言为 JPDL，使用 JPDL 提供的流程定义图形化工具设计符合 JBPM 解析规范的流程定义文件。通过 JBPM 提供的流程部署接口，将流程定义文件部署到 workflow 引擎中。根据需求分析的描述，workflow 管理的主要功能就是定义、部署和维护施工管理中的 workflow，其主要功能如下：

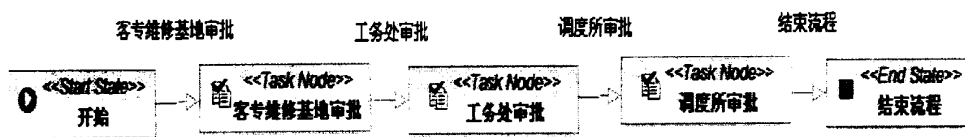


图 4-14 施工日计划流程图

(1)流程定义：使用 JPDL 流程设计工具来定义业务需要的流程。JPDL 是构建在 JBPM workflow 引擎上的流程定义语言，它是基于 Eclipse 的插件，

通过可视化的方式定义流程，直观地描述业务流程。可视化方式的流程定义只能帮助我们完成有限的流程定义设计，很多 JPD L 语言的高级特性和复杂流程的完成都需要结合 JPD L 定义文件的 XML 源代码进行实现。

(2)部署流程：将定义好的流程部署到施工管理子系统中，以便在使用流程的时候可以选择流程。用户将定义好的流程定义文件和图片上传到系统中，系统首先将流程定义的 XML 文件通过 JBPM 提供的部署接口解析，然后向 JBPM 流程定义表 JBPM_PROCESSDEFINITION 中添加新的流程定义记录，同时向其子表添加其他节点等详细信息，最后把流程名称、定义文件和定义图片等流程信息保存至工作流程管理的数据表中。

(3)查看流程：用于查看流程定义信息。从 workflow 表中读取已部署的流程数据并以下拉框的方式根据流程名显示出相应的流程定义文件和流程图。

(4)删除流程：用于删除已经部署到系统中的流程，当用户点击删除流程的时候，将删除 JBPM 工作流引擎中的流程部署和公文管理子系统中保存的流程信息。其中，由于 JBPM 的版本管理机制，需要删除部署在 JBPM 工作流引擎中的所有同名流程定义。

2. 施工日计划上报审批管理设计

通过第三章对施工计划的需求分析，施工日计划上报审批管理有 4 个菜单。

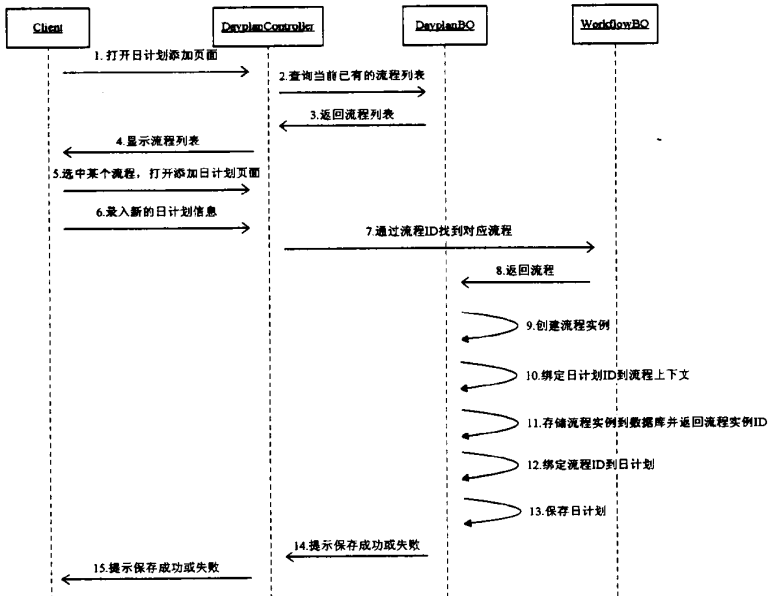


图 4-15 添加日计划时序图

(1)日计划列表：用于查询、更新、删除用户创建的日计划信息和添加日计划。从日计划表中读取当前用户创建的日计划并以列表形式显示，列表显示的信息包括单位名称、线名、行别、起始车站、终止车站、施工日期、创建日期、流程状态、附件、审批历史。日计划只有在未进入工作流流转时才能更新或删除。添加日计划的时序图如图 4-15。

(2)待审日计划列表：待审日计划是指需要当前登录用户审批的日计划，以列表形式显示从工作流引擎查找的当前用户任务列表中对应的计划信息。用户任务包括被制定到用户本人的任务以及用户所在角色组的任务。当用户填写审批信息以后，点击提交到下一步流程。日计划将从待审日计划列表移除而进入已审日计划列表中。审批结束后，公文状态修改为当前节点返回。提交日计划时序图 4-16 如下：

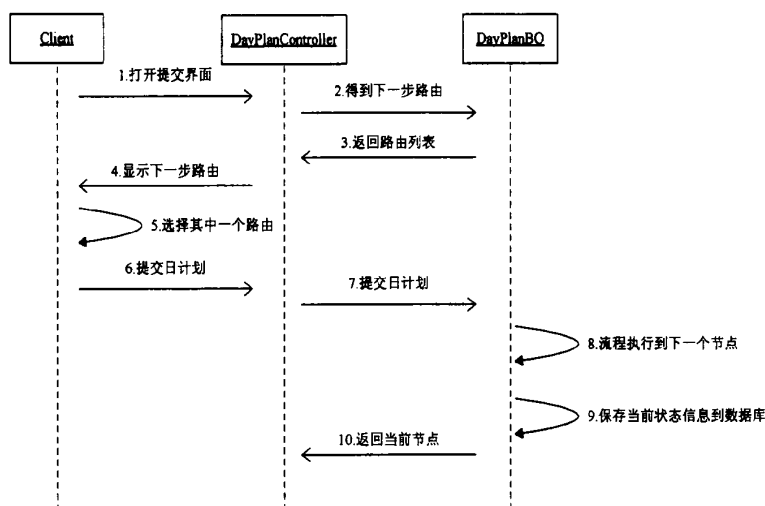


图 4-16 提交日计划时序图

(3)已审日计划列表：显示用户已经审批过的日计划，从计划审批历史表中读取用户审核过的日计划并以列表的形式显示在页面上。列表显示的信息有：计划号、单位名称、线名、行别、起始车站、终止车站、施工日期、创建日期、流程状态、附件、审批历史。

4.4 系统数据库设计

4.4.1 数据库概念模型设计

数据模型是现实世界中数据特征的抽象^{[36] [37]}。概念数据模型也称信息

模型，它以实体-联系(Entity-RelationShip,简称 E-R)理论为基础，并对这一理论进行了扩充。它从用户的观点出发对信息进行建模，主要用于数据库的概念级设计^[38]。下面以编制施工项目台账、施工组织设计和计划流程为例，设计数据库概念模型。

从第三章需求分析可以看出，项目台账和施工组织设计的编制者和使用者主要包括维修基地用户、施工单位清筛车间和工区用户、施工单位维修车间和工区用户。现在把他们合并为一个实体“用户”，不同用户可以使用功能由角色和权限来控制。从项目台账和施工组织设计的业务中可以得出主要实体的关系图如下：

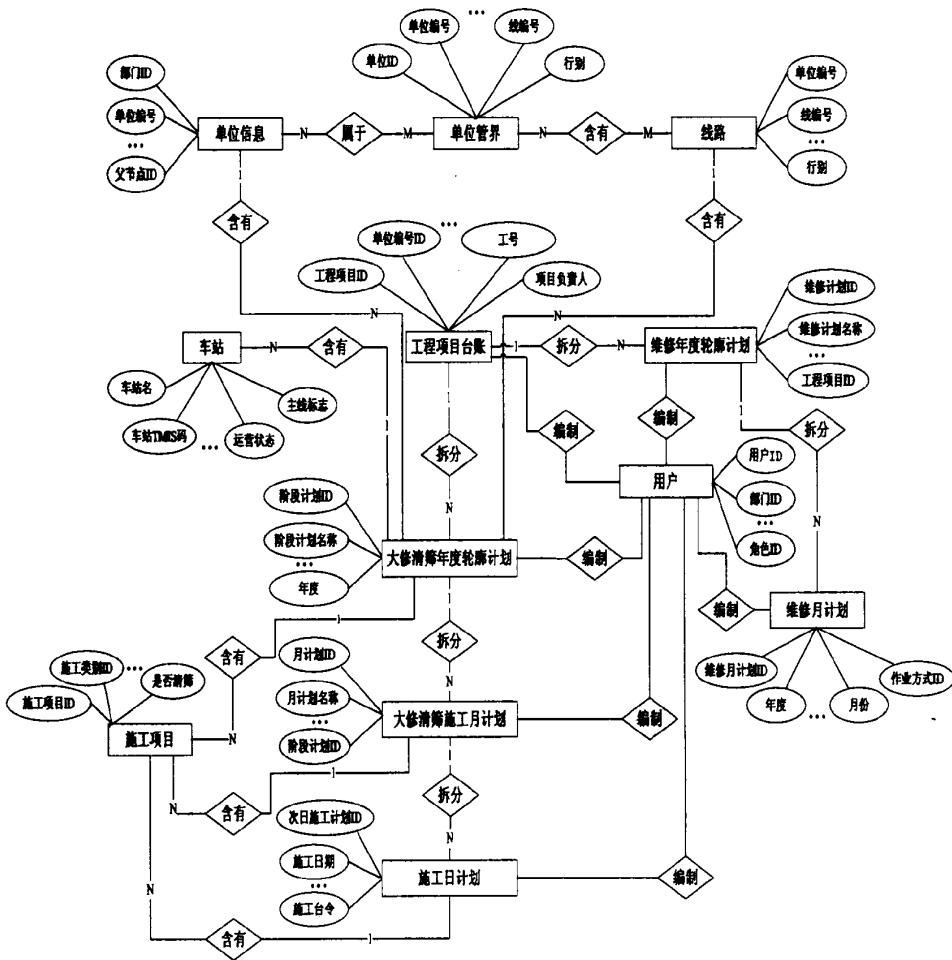


图 4-17 系统局部 E-R 图

4.4.2 数据库逻辑结构设计

基于对施工管理子系统的概念模型设计，将基本的 E-R 图转换为相应的逻辑结构，设计相关的数据表。根据概念模型的设计，施工管理子系统中主

要使用的数据库表如下：

表 4-1 主要数据库表描述

序号	数据库表	功能描述
1	D_SG_ConstructProject	工程项目台账
2	D_SG_ConstructPhasePlanMain	大修清筛施工年度轮廓计划主表
3	D_SG_ConstructPhasePlanItem	大修清筛施工年度轮廓计划次表
4	D_SG_MaintenancePlan	维修年度轮廓计划
5	D_SG_ConstructMonthPlanMain	大修清筛施工月计划主表
6	D_SG_ConstructMonthPlanMain	大修清筛施工月计划次表
7	D_SG_MaintenancePlanPerMonth	维修月计划
8	D_SG_ConstructNextDayPlanMain	次日施工计划主表
9	D_SG_ConstructNextDayPlanItem	次日计划施工项目表
10	D_SG_ConstructPractice	施工情况表
11	D_SG_SpeedLimitedZoneRecord	营业线限速地段表
12	D_SG_FangSanRecord	放散登记表
13	D_SG_UnloadingRecord	路料卸车登记表
14	D_SG_RailCarPlanApply	轨道车运行计划
15	D_SG_CarTurnaroundPlanApply	车辆、宿营车转场计划
16	D_SG_EquipmentStationTrack	大机设备驻地追踪
17	D_SG_UnloadDaoChaApply	卸车（风动碴、轨枕、高边碴）计划
18	D_SG_ConstructPracticeCheck	施工验收情况表
19	D_SG_PATHINFORMATION	轨顶高程资料
20	D_SG_CURVEINFORMATION	曲线竣工资料
21	D_SG_COMPLFILE	竣工文件管理
22	D_SG_OperationRevealed	运行揭示
23	C_ConstructItem	施工项目字典表
24	C_ConstructUnit	施工单位字典表
25	C_ConstructLeader	施工负责人字典表
26	D_SG_ConstructDayPlan	正式施工日计划
27	D_SG_ConstructAcceptanceCheck	施工交验表
28	C_Workflow	工作流表
29	C_ApproveInfo	审批信息表

在表 4-1 中，最主要的表是工程项目台账表 (D_ConstructProject)，它是其他各类计划制定的基础，表 4-2 是它的表结构设计。

表 4-2 工程项目台账表结构

序号	字段名	类型	字段解释
1	GCMID	Varchar(20)	工程项目 ID,主键, not null
2	DEP_ID	Varchar(20)	单位编号, not null
3	GCMC	Varchar(500)	工程项目名称, not null
4	GONGHAO	Varchar(500)	工号
5	XMFZR	Varchar(100)	项目负责人
6	XMFZRSJ	Varchar(11)	项目负责人手机
7	JHZGZL	Number(10,3)	计划总工作量
8	JHZGQ	Integer	计划总工期
9	JHZGS	Integer	计划总工时
10	YSZE	Number(20,3)	预算总额
11	SJKGRQ	Date	实际开工日期
12	SJWGRQ	Date	实际完工日期
13	CZRID	Integer	操作人 ID
14	CZSJ	Date	操作时间
15	REMARK	Varchar(500)	备注
16	SCBZ	Integer default 0	删除标志
17	SGLBID	Varchar(2)	施工类别 ID
18	SGQSRQ	Date	施工起始日期
19	SGZZRQ	Date	施工终止日期

4.5 权限管理平台设计

4.5.1 权限管理平台设计目标

设计一个灵活、通用、方便的权限管理平台，该平台不仅要满足“客专维修基地综合维修信息系统”对权限管理的特定需求，而且在此基础上抽象、提取出平台所需的通用标准数据接口，使之可以为各种业务应用系统提供统一的权限管理功能。

对于业务系统来说，它并不关心权限管理平台是如何进行权限管理的，故对于业务系统而言，本平台为一可根据业务定制的中间层，在输入端输入必要的信息（用户名、密码、功能模块编码等），在输出端反馈所查询权限状态。简言之，本平台的目标是：为业务系统提供一个应答在定制体系中特定用户是否拥有特定权限的功能平台。

4.5.2 管理对象

➤ 权限管理

所谓权限管理指对系统资源（如数据库对象、网页等）的访问权和使用权的管理。

➤ 数据审计

对用户设定的敏感数据进行监控，记录用户对此类数据的操作行为，进一步实现数据向特定时间点（SavePoint）回滚。

4.5.3 权限控制模型选定

访问控制技术是由美国国防部资助的研究和开发成果演变而来的。这一研究导致两种基本类型访问控制的产生：自主访问控制(Discretionary Access Control, DAC)和强制访问控制(Mandatory Access Control, MAC)^[39]。最初的研究和应用主要是为了防止机密信息被未经授权者访问，近期的应用主要是把这些策略应用到商业领域。

DAC 它能够控制主体对客体的直接访问，但是不能控制主体对客体的间接访问，例如利用访问的传递性，a 能访问 b，b 能访问 c，于是 a 也可以访问 c。可见这种由授权定义的存取限制很容易被旁路，使系统无法对抗对数据库的恶意攻击。虽然这种自主性为用户提供了很大的灵活性，但同时也带来了一些安全问题^[40]。

MAC 主要用于多层次安全级别的应用中，预先定义用户的可信任级别及信息的敏感程度安全级别，当用户提出访问请求时，系统对两者进行比较以确定访问是否合法。安全级别较高的计算机采用这种策略，常用于军队和国家安全机构。其缺点在于主体访问级别和客体安全级别的划分与现实要求无法一致，在同级别间缺乏控制机制。访问级别的划分不够细致。另外 MAC 由于过于偏重保密性，对其他方面如系统连续工作能力、授权的可管理性等

考虑不足^[41]。

当前研究最广泛的模型是基于角色的访问控制 (Role Based Access Control, 简称 RBAC)^[42], 该模型最早是由美国 George Mason 大学的 Ravi.Sandhu 在 1996 年提出的。RBAC 最显著的特点之一就是针对用户相对变化较多、角色相对变化较少的实际情况, 使角色关系具有相对稳定性和易维护性, 当系统权限发生变化时, 只要重新对角色授权, 就能实现其对应用户的权限改变^[43] ^[44]。这种简洁的思路使系统的授权方式更加灵活, 而且极大的缓解了授权管理的负担。RBAC 模型如图 4-1 所示。

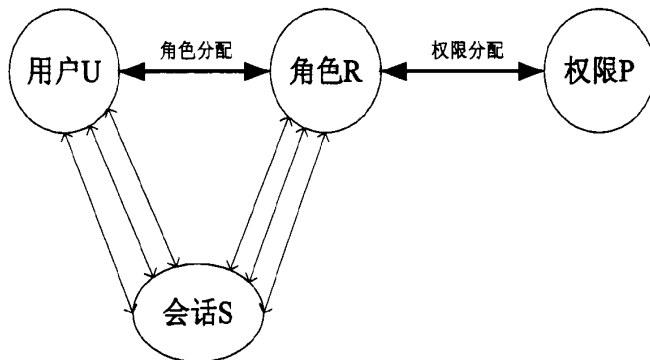


图 4-18 基于角色的访问控制模型 RBAC96

RBAC 模型定义了四个实体和两个分配关系:

(1)用户 U (Users): 指一个可以独立访问计算机系统中的数据或用数据表示的其他资源的主体。用户在一般情况下是指人, 但有时也包括计算机或一些组织。

(2)角色 R (Roles): 指执行特定任务的能力或在组织中已被授予一定责任的工作头衔。它代表一种资格、权利和责任。

(3)权限 P (Privileges): 在系统中对一个或多个客体进行特定模式访问的操作权限, 与实现机制密切相关。权限的本质取决于系统的实现细节, 如操作系统中保护的是文件、目录等资源, 相应的操作是读、写、执行等。

(4)会话 S (Sessions): 每一个用户在系统执行时, 可以建立数个会话, 而每一个会话又可对应数个角色, 或称每个会话都拥有一个“活跃角色集”ARS (Active Role Set)。这样用户便可以动态选择所需角色, 仅使用所需权限, 从而可实现最小权限原则。

(5)用户分配 UA (User Assignment): 表示用户角色分配关系, 是一个多对多的映射关系。

(6)权限分配 PA (Privilege Assignment): 表示角色权限分配关系, 也是一个多对多的映射关系。

4.5.4 权限平台的授权机制

RBAC 模型是抽象、普适的模型，距离我国铁路信息化建设业务需求尚有一定的距离，为了构建一套通用的权限管理平台，实现对资源的统一访问控制，需要对 RBAC 模型进行裁剪和改进，设计出一种符合我国铁路信息化建设需求的 RBAC 实现模型。

本平台拟采用“角色-用户”（RU）+“用户-权限”（UP）的设计思路，简称 RUP 模型，如图 所示。可以看到本方案定义了三个实体以及三个分配关系，相对 RBAC（见图 4-1），RUP 模型增添了一个“用户权限分配”方式，这弥补了 RBAC 模型在实际应用中的几个不足：

(1)当角色不存在时，系统管理员同样可以给用户添加权限。根据实际情况，这种情况比较常见，如系统刚建成时，使用单位对于角色的划分尚不清晰；或仅需要给用户分配一些简单权限时，不需要添加一个角色的动作，简化了用户操作，提高用户体验；

(2)对于某些权限，并不对应一个明确的角色概念，或需要临时赋予用户一些权限时，如果这类权限也需要同一个角色挂钩，将会导致系统角色过多、角色责任不明确的问题。因此，这种情况应当提供直接将用户同权限关联的功能。

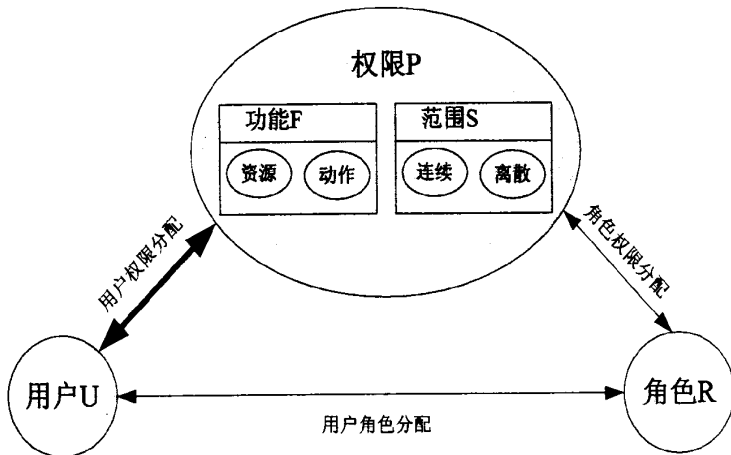


图 4-19 基于“角色-用户-权限”的访问控制模型 RUP

RUP 模型的基本原理：在 RUP 模型中，权限 P 由功能 F 与范围 S 构成，系统管理员可以直接把权限赋予用户；同时，为了便于对各个独立权限的管理，可以采用角色 R 将权限进行归类。很明显，RUP 模型弱化了角色的概念，支持用户直接被赋予权限（如图 所示，对“用户-权限”的关系线进行了加粗），并提供了用户与角色的对应关系，增强了系统的灵活性。

RUP 模型的概念解析:

(1)功能 F: 指在系统中完成业务单元动作所需的资源访问权。其中资源指数据表、文件等系统元素,动作指增删改查等;

(2)范围 S: 指对资源访问的范围,是对功能的限制。在本平台中,我们设计了两类范围描述方式:连续范围、离散范围。举例:

连续范围:用户 U1 属于某工务段,他可以访问的数据应该在其工务段所在管界范围内,如京沪线 150km~300km,这可以用连续范围描述;

离散范围:某一资源只能由高工、主任级别的用户访问,则可以通过离散范围描述;

(1)权限 P: 由功能和范围组成,明确指定对一定范围内特定资源的访问方式,如:对京沪线 150km~300km 范围内线路检测信息的添加操作;

(2)角色 R: 权限的集合,可以直接赋予用户;

(3)用户 U: 同 RBAC 模型,指可以独立访问计算机系统中的数据或用数据表示的其他资源的主体。

4.5.5 权限平台系统设计

整个权限管理平台由两部分组成,如图 4-20 所示。

(1)管理控制台:根据 RUP 模型原理进行权限管理操作;

(2)应用程序调用控制台

它是一组共用模块构件,供业务系统各应用模块调用,根据权限管理模型完成对用户数据库数据的读写权限控制。包括:(a)用户身份确认检查;(b)权限控制服务:业务系统提出请求,根据用户身份确认检查提供的用户所属角色名,通过查询数据库取得该用户所拥有的功能权限,由权限管理平台返回是否可执行的指令;(c)会话信息管理:用户登录后的会话信息、操作信息管理;(d)用户视图控制:根据权限控制理论的“最小权限”原则,根据用户身份,为其自动定制相应的视图,既满足用户的功能需求,且无法越权操作。

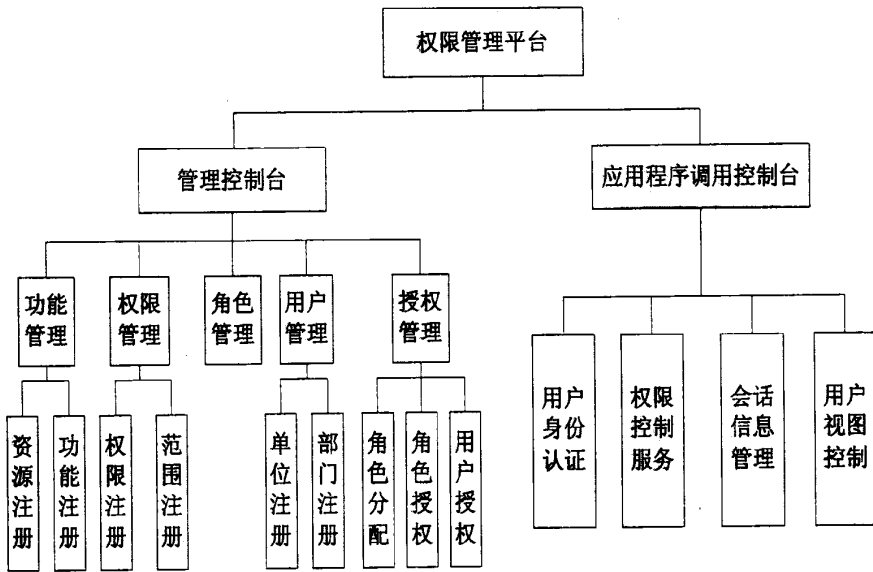


图 4-20 权限管理平台功能结构

4.5.6 权限平台系统架构

权限管理平台系统架构图如 4-21 所示。外部业务系统通过用户认证接口或应访问接口访问内部权限管理系统。权限管理模块根据接口数据信息对数据库进行查询，并将查询结果返回给业务系统。通用权限管理系统的体系结构实现了以用户为中心，并确保用户安全可靠地访问相应的功能和信息。

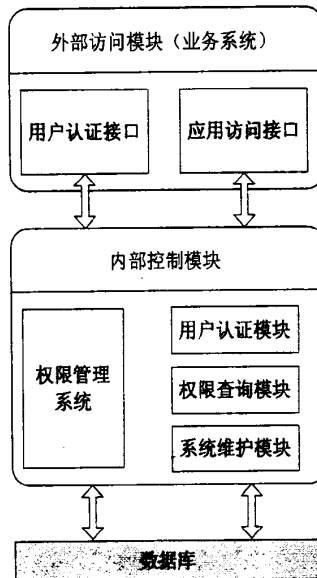


图 4-21 权限管理平台系统架构图

4.5.7 RUP 模型设计

(1)重要关系表

➤ 用户权限分配表

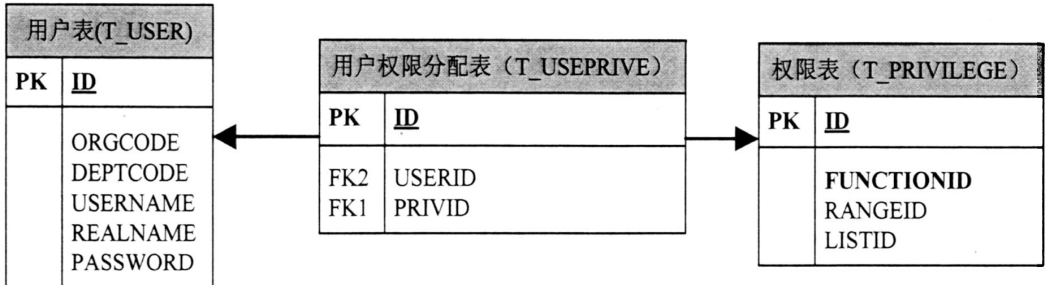


图 4-22 用户权限分配表

本关系表主要将用户表与权限表进行关联。

➤ 角色权限分配表

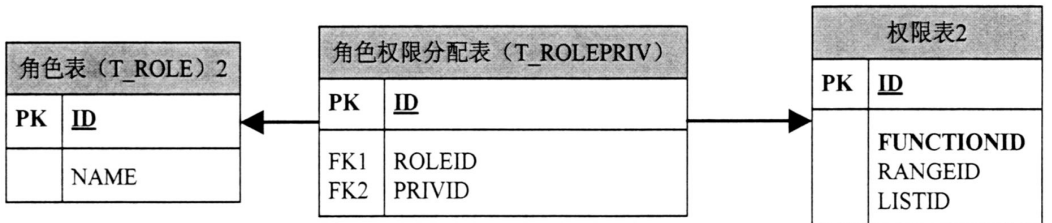


图 4-23 角色权限分配表

本关系表主要将角色表与权限表进行关联。

➤ 用户角色分配表

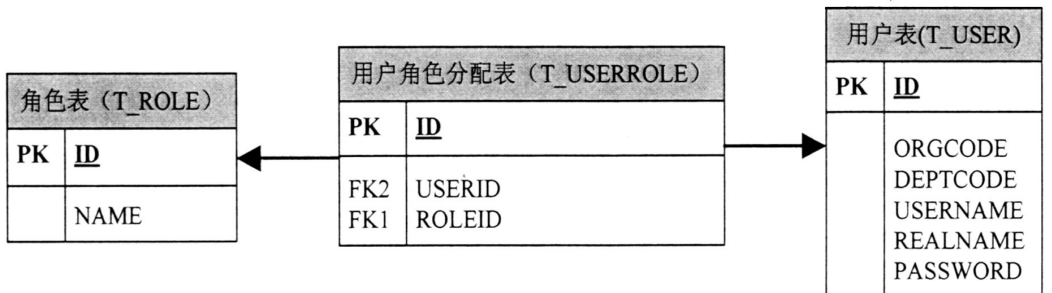


图 4-24 用户角色分配表

本关系表主要将用户表与角色表进行关联。

(2)权限管理平台数据库设计总图

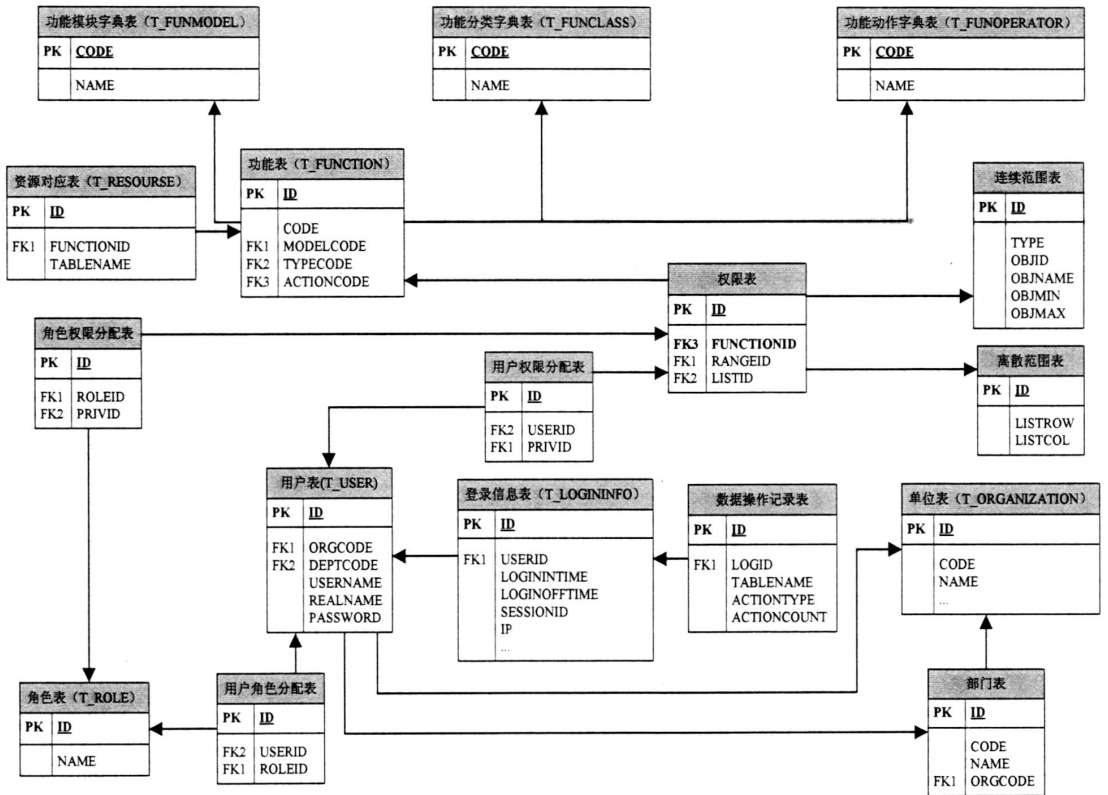


图 4-25 权限管理平台数据库设计总图

4.5.8 其他相关设计

(1) 分级授权

如果用户拥有“授权”角色，则该用户还可为其他用户授权，这种授权可以是分级进行的，被授权者的权限范围不能超过授权者的权限范围，具体做法是：在用户表中添加“授权者”字段，可以表明该用户之权限继承自该授权者，从而使该用户的权限范围控制在授权者的权限范围之内，从而实现分级授权。每个用户都有一个授权者，顶级用户授权者为系统管理员。

(2) 权限制约

由于用户和角色之间是“多对多”的关系，一个用户可以拥有多个角色，从而自然地解决了角色的灵活配置与控制。例如：将“授权”和“审计”两个角色同时赋予同一个用户，则该用户同时拥有授权和审计两种权限，此时实现了简化操作；反之，也可以将“授权”和“审计”两个角色分别授予不同的用户，此时实现了权限制约。

(3) 权限审计

有了权限制约的结构以及两权（“授权”与“审计”）分立的体系，为权

限审计提供了可能性。权限审计包含两方面含义，第一，授权后权限并不立即生效，需要审计；第二，权限使用时必须先行校验权限的有效性。具体做法是：在权限分配表和角色分配表中添加“是否启用”字段，授权操作在权限表中新增记录时该字段默认为“否”，经拥有“审计”角色的用户确认后将该字段值置为“是”，权限方能生效，从而实现权限审计。

第 5 章 施工管理子系统实现与测试

5.1 系统的实现环境

系统的实现环境如表 5-1 所示：

表 5-1 系统实现环境表

开发工具	JDK1.6、Weblogic workshop10.3g、Eclipse3.2
应用服务器	PC 服务器或刀片服务器, 操作系统为 Windows 2003 Server
数据库服务器	Oracle 10 服务器版, 操作系统为 unix
应用平台	Weblogic10 为开发平台, Weblogic Server10 为应用服务平台
开发框架	Struts 1.2 、Beehive 1.0.1、 Hibernate3.0
工作流	Jbpm 3.2.3
网络环境	铁路内部网络
浏览器	IE 6.0

5.2 施工管理子系统实现

1. 域模型层的实现

通常编码都是从底层开始的，数据访问层继承了 JdbcControl 控件，控件实现了 Java 类和数据库表的映射，和数据库的链接、发送请求与返回结果集以及数据库事务。我们在开发时可以方便的操作 Java 类也就是域模型。域模型主要包括一个与表对应的 VO 值对象（也即 PO，持久化对象）。

这里仍然以施工组织设计编制为例，PhasePlanVO.java，其部分代码如下，主要提供一组 Setter/getter 方法，并在缺省构造方法中设置该 VO 对象所对应表的表名、主键、查询所有结果集的存储过程名。这三个值是作为参数供 BaseBO 逻辑层中方法调用。

表 5-2 PhasePlanVO 类结构

```
Public Class PhasePlanVO implements Serializable{
    Public PhasePlanVO(){
        this.pk = "JDJHID";
        this.tableName = "D_SG_ConstructPhasePlanMain";
        this.functionName = "djyy.SCGL_SGGL.getPhasePlansByCase";
    }
}
```

```
}  
    Private String jdjhid;  
    Private String dep_id;  
    .....
```

需要说明的是 VO 类中的字段和数据库表字段要按照顺序一一对应。域模型的一个重要用途是实现粗粒度的数据传递方式，以保持接口的相对稳定。我们把对象 PhasePlanVO 作为 FormBean 的一个属性，以接收客户端的数据录入，并作为一个整体在 Controller 以及 BO 各层之间传递数据。

2. 数据访问层的实现

数据层由 ReadControl 数据库控件和 WriteControl 数据库控件组成。这两个控件均实现了 JdbcControls 控件接口，JdbcControls 控件是系统控件，使用系统控件可以连接到常用的应用程序资源。如数据库、EJB、JMS 队列、WebService 等，这些控件只需略微修改或不进行修改就可以在应用程序中使用。

JdbcControls 控件封装了数据库访问的具体细节，我们可以通过 JdbcControls 控件向数据库发送任何 SQL 命令，这样便可以检索数据，执行插入和更新操作等。只需采用注释声明式设置的 jndiName 参数和控件方法就可以实现控件的实例化，然后 BO 逻辑层就可以通过数据库控件从数据库检索出数据供页面流客户端 Controller 使用。WriteControl 示例方法如表：

表 5-3 WriteControl 控件结构

```
@ControlExtension  
@JdbcControl.ConnectionDataSource(jndiName = "zhwx")  
public interface WriteControl extends JdbcControl {  
    @JdbcControl.SQL(statement="{sql: sqlStatement}")  
    int update(String sqlStatement) throws SQLException;  
    .....
```

3. 业务层的实现

业务层主要根据业务的需要，对 BaseBOImpl 类的方法返回的数值或者其他相关联的业务逻辑类的方法返回的数值进行处理，并根据需要调用工具类辅助业务的处理，将处理的数据结果供表示层使用。

业务层的编写采用自定义控件，创建自定义控件的好处是可以全面定义对资源的访问或封装某些应用程序功能。可以设计一个自定义控件来完成应

用程序中的任何任务。施工组织设计编制的业务层由 ConstructProjectBO、GjglBO、ShareUserBO 三个自定义控件接口类及其实现类组成，ConstructProjectBO 控件接口主要提供对施工项目台账的动态信息操作方法。GjglBO 控件接口提供对工程管理相关的动态信息操作方法，如获取线名、线编号、行别、车站、下一级单位列表等。ShareUserBO 控件接口提供查询用户和根据用户查询部门信息等方法。ConstructProjectBO 的示例方法如下所示：

表 5-4 ConstructProjectBO 控件接口结构

```
@ControlInterface
public interface ConstructProjectBO {
    void insertConstructProject (ConstructProjectVO constructProjectVO
        ) throws Exception;           //添加工程情况
    void updateConstructProject (ConstructProjectVO constructProjectVO
        ,String PK) throws Exception; //更改工程情况
    public Date getKaiGongTime(String gcxmid) throws Exception;
        //根据工程项目 ID 获取开工时间
    public String[] getWxjhid(String gcxmid) throws Exception;
        //根据工程项目 ID 获取维修计划 ID
    public int getNextDayPlanscount(String strb) throws Exception;
        //按条件取得次日施工计划情况
```

4. 表示层的实现

表示层由控制层和视图层组成。控制层应用 JPF (java page flow) 页面流技术，视图层主要应用 NetUI 标记库。JPF 基于 Struts1.1 构建，但省略了诸多繁琐的工作，主要省略了部署配置文件的管理，也无需为每一个业务单独编写一个 Action 以及对应的 ActionForm 类。JPF 由一组页面和一个注释驱动的控制类来组成应用，控制类里包含了动作方法 Action。Action 可以处理浏览器的要求以及复杂的功能，其返回值为 Forward，跳转方法由 @Jpf.Forward 注释来描述。

施工组织设计编制的 JPF 由控制类 DesignController 和页面 SgzsjDownloadDesign.jsp、SgzsjQueryDesign.jsp 以及动作 begin, SelectAction,download_Action,filedown_Action,fileUpload_Action,以及 selectAll_Action 组成。这些页面和动作方法的作用如下表所示。

表 5-5 DesignController 类中的动作与页面

SgzszjQueryDesign.jsp	分页展示施工项目查询列表
SgzszjDownloadDesign.jsp	项目对应的文件上传下载页面
begin	JPF 默认的初始动作
SelectAction	检索所有的项目信息
download_Action	进入上传下载页面前动作方法
filedown_Action	下载文件
fileUpload_Action	上传文件
selectAll_Action	根据查询条件检索项目信息

它们之间的先后调用关系如图 5-1 页面流概览图所示：

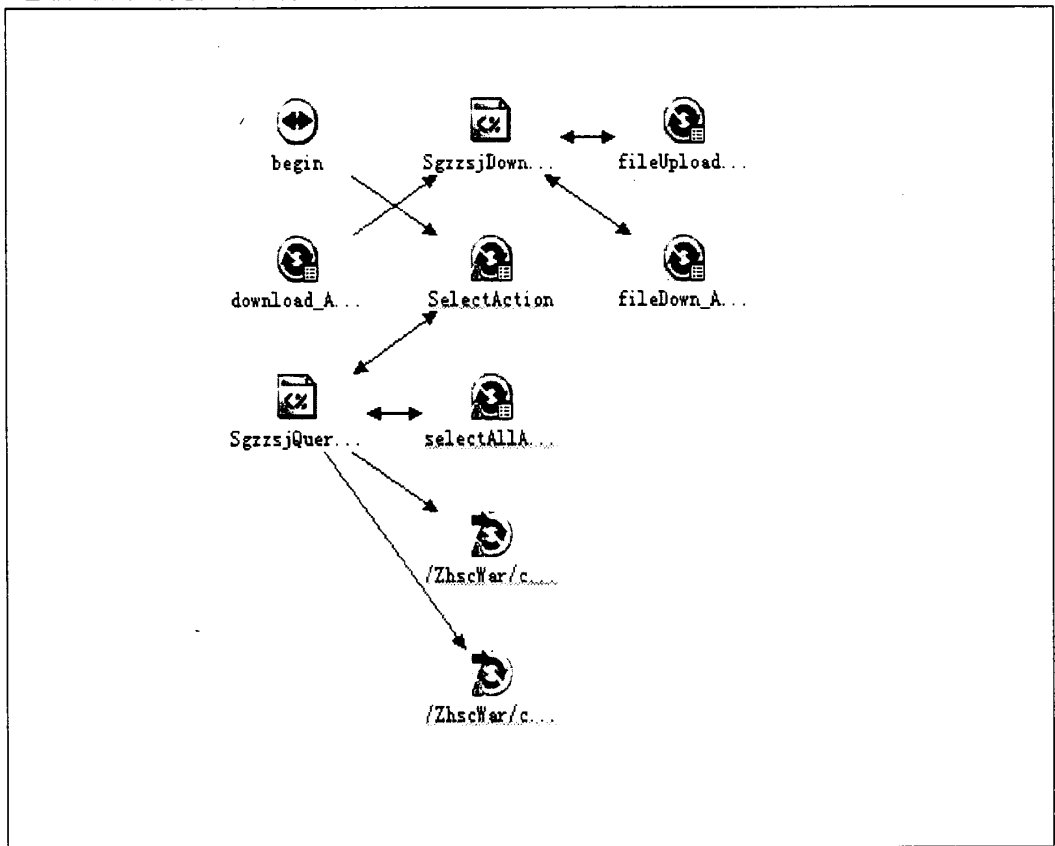


图 5-1 施工组织设计编制页面流概览图

视图层主要就是 JSP 页面。由 JSP 标签、NetUI 标签、JSTL 标签组成，还利用了 AJAX 的相关技术，利用 CSS(Cascading Style Sheet, 级联样式表)对页面进行修饰，力求达到界面统一和美观。

NetUI 通过三个标签文件提供标签，tags-html-1.0 标签库中封装了各种 html 元素，tags-databinding-1.0 标签库中提供了数据绑定功能，并由 dataGrid

和 `repeater` 标签来显示表格和列表。Tags-template-1.0 标签库提供页面模板的功能，能更有效去组织页面。

对视图层的设计主要体现在页面表单的设计和列表数据的设计，主要应用 Dreamweaver 先进行页面的设计，具体效果再由 NetUI 设计。

5.3 系统关键性技术的实现

5.3.1 报表导入的实现

由于各单位运行揭示报表格式统一，均按照一定格式以 Excel 文档填报，在运行揭示管理模块中考虑直接导入 Excel 文件到数据库中。比较常用的导入 Excel 文件的技术方案有 JXL 技术和 POI 技术。

POI 是 Apache 基金会下面的一个子项目。它的功能很强大，支持对 Excel 的处理是其中的一部分功能，但是它对中文支持不太好。Jxl 是韩国的软件工程师写的 java 操作 Excel 的工具，优点是对中文支持非常好，并且 API 是纯 Java 的，并不依赖 Windows 系统，即使运行在 Linux 下同样能够正确的处理 Excel 文件。且功能操作也比较强大。Java 开发人员可以通过它读取 Excel 文件的内容、创建新的 Excel 文件、更新已经存在的 Excel 文件，并可以对文件内容进行格式化、非常灵活的操作 Excel 文件的行和列、插入图片等。在 JVM 内存消耗方面，JXL 消耗更小，导入效率更高。

根据我们导入数据的需要，没有必要使用功能强大的 POI，使用能满足导入功能需求的 JXL 即可。JXL 操作 Excel 文件的角度可以这样理解 Excel 文件结构：每个 Excel 文件就是一个 Workbook 对象 `book`，每个 `book` 包含多个 Sheet 对象，每个 Sheet 中的单元格就是 Cell 对象，每个 Cell 对象的实例都有行、列和值三个属性。基于以上的分析，系统中，我们在运行揭示控制类 `xjsController` 中设计了 Excel 文件导入方法 `insertYxjsAction`，导入 Excel 文件的步骤大概分成以下几步：

(1) 将上传文件保存到服务器，路径选择在服务器 D 盘 `yxjs` 文件夹下，再从本地文件从读取 Workbook 对象 `wb`。

(2) 从 `wb` 中读取工作表 Sheet 对象 `rs`。

(3) 在两层循环的 `rs` 循环体内，根据行、列的变化读取单元格的值并保持到 `jxjsVO` 对象中。

(4) 将 `jxjsVO` 对象保存到数据库中。

图 5-2 是运行揭示导入页面图：



图5-2 运行揭示文件导入图

5.3.2 页面功能点权限控制

系统安全性是衡量一个系统是否优秀的重要指标。权限控制完成的是允许登录后的用户做什么事。出于安全性的考虑，在施工管理作业过程中，各级登录用户只能在各自权限允许的范围内进行添加、修改等动作，不能越权进行操作。

这就涉及页面功能点权限控制技术的实现，由于用户角色和其拥有的功能模块的操作权限是一一对应的，在用户登录时候系统根据用户角色查询出所有操作访问权限并保存在 Session 中，当用户进入到每个功能页面时，通过 C 标签语句判断该用户拥有的权限来决定是否提供操作功能。以查询和增加施工组织设计编制为例：

- ▶ 登陆页面后系统将操作访问权限保存在 Session 部分代码

表 5-6 登陆用户获取各功能操作权限

```

RoleFuncVO[] funcs = roleBO.getRoleFuncByID(Integer.valueOf
(user.getROLEID()));
if (funcs != null && funcs.length > 0) {
    for (int i = 0; i < funcs.length; i++) {
        if BigDecimal.valueOf(0).equals(funcs[i].getAUTHVALUE()) {
            user.setPrivString(user.getPrivString() + "," + funcs[i]
                .getFUNCCODE());
        }
    }
    UserPrivs privs = new UserPrivs();
    privs = VOUtil.setUserPrivs(privs, user.getPrivString());
    session.setAttribute("user_priv", privs);
    .....

```

用户登录功能页面后系统对是否具有查询和添加访问权限的判断

表 5-7 判断用户是否具有权限的 C 标签

```
<c:if test="${user_priv.DJYY_SG_GCXM_QUERY==0}">
  <netui:button styleClass="b_chaxun" type="button" value="查询" onClick=
    "chaxun()"/>
</c:if>
<c:if test="${user_priv.DJYY_SG_GCXM_ADD==0}">
  <netui:button styleClass="b_tianjia" type="button" value="添加" onClick=
    "add()" />
</c:if>
```

5.3.3 workflow 关键部分实现

(1) 创建流程实例

根据选择的工作流程创建施工日计划的流程实例，返回该实例的 ID，并且将施工日计划 ID 通过变量设置在上下文中。此实例控制施工日计划的流转。

表 5-8 日计划流程实例的创建

```
public long addProcessInstance(String wf_name, int dpId) {
    JbpmContext context = getContext();
    ProcessDefinition pd = context.getGraphSession().findLatestProcessDefinition
        (wf_name);    // 通过流程名称得到流程定义
    ProcessInstance pe = new ProcessInstance(pd);
    //通过流程定义取得流程实例
    pe.getContextInstance().createVariable("dayplan", dpId);
    // 将日计划的 ID 通过变量设置在上下文中
    context.save(pe);    // 存储流程实例到数据库中
    context.close();
    return pe.getId();
}
```

(2) 将日计划添加进选定的流程，绑定日计划和流程

将流程设置到日计划中，通过创建流程实例函数得到新创建的流程实例的 ID，将流程实例 ID 设置到日计划中。这样通过流程实例 ID 和日计划 ID 的互相设置完成日计划和流程的绑定。

表 5-9 日计划、流程绑定

```
public void addDayPlanForProcess(DayPlan dp, WorkFlow wf) {  
    dp.setWorkflow(wf); //设定日计划的流程  
    save(dp);  
    long processInstanceId = addProcessInstance(wf.getPlanname(),dp.getPid());  
    dp.setProcessInstanceId(processInstanceId);  
    //将流程实例 ID 设置到日计划中  
    update(dp);  
    //更新日计划  
}
```

(3)任务流转

任务流转是流程实例按照流程定义到达下一个任务。Jbpm 开始流程和提交下一步流程调用方法不同。在任务流转中需要分别进行处理。如果当前节点是起始节点，就调用 Signal 方法进入到流程，Signal 方法在一个实例中只会被执行一次。如果当前节点是非起始节点，那么调用 end 方法进入下一步流程，end 方法首先会结束任务实例对象，然后触发当前实例对象的 Signal 方法，这样使得流程继续往下流转。

表 5-10 任务流转代码

```
private String getStatus(String uname, long processInstanceId, String  
    transitionName) {  
    JbpmContext context = getContext();  
    String status = null;  
    ProcessInstance pi = context.getProcessInstance(processInstanceId);  
    //得到流程实例  
    String currentNodeName = pi.getRootToken().getNode().getName();  
    //得到当前的节点  
    String startNodeName = pi.getProcessDefinition().getStartState().getName();  
    //得到起始节点  
    if(currentNodeName.equals(startNodeName)){  
        pi.getRootToken().signal(transitionName);  
    }  
}
```

```

}else {
    List<TaskInstance> tasks = context.getTaskMgmtSession()
        .findTaskInstances(uname); //找出当前用户的当前任务
    for(TaskInstance task:tasks){
        if(task.getProcessInstance().getId() == processInstanceId){
            task.end(transitionName);
            break;
        }
    }
}
status = pi.getRootToken().getNode().getName();
// 将日计划当前所处的节点作为状态信息返回
if(pi.hasEnded()){ // 判断当前状态是否结束
    status = DayPlan.STATUS_END;
}
return status;
}
}

```

(4)复杂流程定义

JPDL 提供的可视化流程插件可以很容易实现流程定义，如图所示的施工日计划申请审批流程。但是可视化的流程定义工具并不能完成流程定义的任务实例分配给多个参与人的功能，如日计划流程中的维修基地审批，由于维修基地可能存在多个审批人，流程实例并不清楚当前维修基地节点谁来负责审批，流程定义文件更不能将处理这个流程的工务处主任名称固定，因此需要对流程定义文件进行编辑来完成复杂流程定义。

施工日计划流程中实现了 Jbpm 提供的 AssignedHandler 接口处理流程定义中，其流程定义代码如下：

表 5-11 复杂流程定义

```

<?xml version="1.0" encoding="UTF-8"?>
<process-definition xmlns="urn:jbpm.org:jpd1-3.2" name="日计划审批 2.0">
<start-state name="开始">
    <transition name="客专维修基地审批" to="客专维修基地审批">
    </transition>
</start-state>

```

```
<task-node name="客专维修基地审批">
  <task name="客专维修基地审批">
    <assignment class="com.dj.common.JdAssignmentHandler"></assignment>
  </task>
  <transition name="工务处审批" to="工务处审批"></transition>
</task-node>
<task-node name="工务处审批">
  <task name="工务处审批">
    <assignment actor-id="gwc"></assignment>
  </task>
  <transition name="调度所审批" to="调度所审批"></transition>
</task-node>
<task-node name="调度所审批">
  <task name="调度所审批">
    <assignment actor-id="dds"></assignment>
  </task>
  <transition name="下达给施工车间" to="下达给施工车间"></transition>
</task-node>
<end-state name="下达给施工车间"></end-state>
</process-definition>
```

在上述流程定义的代码中，任务分配由制定的类来完成，这个类需要实现 JBPM 的任务分配接口 `AssignmentHandler`，其实现代码如下：

表 5-12 动态指派参与者代码

```
public class JdAssignmentHandler implements AssignmentHandler {
  public void assign(Assignable assignable, ExecutionContext context) throws
  Exception {
    Connection conn = context.getJbpmContext().getConnection();
    Statement stmt = null;
    ResultSet rs = null;
    try {
      String sql = " select username from djuser where roleid = 2";
      stmt = conn.createStatement();
```

```

rs = stmt.executeQuery(sql);
List ls = new ArrayList();
while(rs.next()){
    ls.add(rs.getString("username"));
}
String[] actorIds = new String[ls.size()];
ls.toArray(actorIds);
assignable.setPooledActors(actorIds);
} finally{
    rs.close();
    stmt.close();
    conn.close();
}
}
}
}

```

5.4 实现界面

5.4.1 施工组织设计页面

施维修基地管理系统

您的位置首页 > 生产管理 > 施工组织设计 > 施工组织设计编制

单位: 武汉维修基地 工程项目名称: 工号:

施工项目台账列表

序号	单位	施工类别	项目名称	工号	总工作量	总工期	总工时	预算总额	开工时间	完工时间	操作
1	武汉维修基地	大修	最新测试	98		29			2011-9-1	2011-10-6	删除
2	武汉维修基地	大修	ttt年度清理计划	000006	100000	3	7	111	2012-6-1		删除
3	武汉维修基地	大修	hxytest1	123		16					删除
4	武汉维修基地	维修	维修施工计划	34256	128333	365	2023	23423	2012-6-2	2012-6-2	删除
5	武汉维修基地	维修	维修年度测试	111		22					删除

共5条[首页][上一页][下一页][尾页][第1页] 转到 页[go]共1页 每页显示10条

图 5-3 施工组织设计页面

5.4.2 施工日计划申报审批页面



The screenshot shows a web interface for managing construction daily plans. At the top, there are navigation buttons: '日计划列表', '新增施工日计划', '部署流程', '查看流程', and '退出'. Below these is a table titled '施工日计划申报审批情况列表' (Construction Daily Plan Declaration Approval Status List). The table has 13 columns: 序号 (Serial Number), 计划号 (Plan Number), 单位名称 (Unit Name), 姓名 (Name), 行别 (Line Type), 作业起始车站 (Start Station), 作业终止车站 (End Station), 施工日期 (Construction Date), 创建时间 (Creation Time), 流程状态 (Process Status), 附件 (Attachments), 审查历史 (Review History), 操作 (Operations), and 提交申请 (Submit Application). The table contains four rows of data.

序号	计划号	单位名称	姓名	行别	作业起始车站	作业终止车站	施工日期	创建时间	流程状态	附件	审查历史	操作	提交申请
15	2232	西南交通大学	京津	上	北京	天津	2012-11-23 00:00:00.0	2012-06-05 14:44:48.0	审核结束	下载	查看	删除	已完成
21	777	武汉基地	武广	下行	汉阳	汉阳	2012-06-02 00:00:00.0	2012-06-08 11:19:20.0	新建	下载	查看	删除	提交
22	000	武汉基地	武广	上行	汉口	大花岭	2012-06-14 00:00:00.0	2012-06-08 11:24:02.0	新建	下载	查看	删除	提交
23	678	武汉基地	武广	下行	武昌	贺胜桥	2012-06-21 00:00:00.0	2012-06-15 10:39:47.0	新建	下载	查看	删除	提交

图 5-4 施工日计划申报审批页面

5.5 施工管理模块测试

5.5.1 测试环境

客户端操作系统: Windows Xp Professional

客户端浏览器: IE6.0

WEB 服务器与数据库服务器操作系统: Windows 2003 Server

WEB 服务器: Weblogic 10.3g

数据库服务器: Oracle 10g

性能测试工具: Jmeter 2.6

5.5.2 系统测试

作为系统开发过程中的重要组成部分,系统测试的目的是确认所开发系统的功能和性能是否符合之前业务需求与设计的目标,这个环节也是系统在投入运行前,对系统的需求分析、设计和实现的复审,是保证系统质量的关键步骤。

系统测试是为了发现错误而执行程序的过程。系统测试在系统生存期中涉及两个阶段:一是单元测试,通常在每一个模块编写完之后就需要做必要的测试。另一个是在单元测试后对系统进行的各种综合测试,这是系统生存期的另一个独立阶段,即测试阶段。系统测试的目的是从假定程序中含有错

误的前提下去测试程序，从而可以发现尽可能多的错误，而不是为了证明程序是“好的”。

一般来说，软件测试过程应按 3 个步骤进行，即模块测试、集成测试、系统测试。信息管理系统都是由若干子系统组成。为了加快测试的速度，提高测试效率，应按照“自底向上”的原则依次进行模块测试、集成测试、系统测试工作。

(1)模块测试是针对软件设计的最小单元程序模块，进行正确性检验的测试工作，并且集中对每一个程序单元进行测试，检查程序模块是否争取地实现了规定的功能。Beehive 框架提供了单元测试控件，可使用 ControlTestCase 基类将控件集成到 Junit^{[47] [48]}测试框架中。Junit 是一个基于 Java 编程的系统进行单元测试的测试框架。ControlTestCase 基类提供控件容器，并帮助通过 @Control 批注以声明方式实例化控件。

(2)集成测试是把已测试过的模块组装起来，主要是对设计相关的软件体系结构的构造进行测试。目标是确保系统运行顺利和应用程序各个模块之间的兼容。测试的重点是各模块间的协调与通信，即重点测试子系统内各模块的接口。

(3)经过模块测试和集成测试后，系统测试是把已经经过确认的软件纳入实际运行环境中，与其他系统成分组合在一起进行测试。需要检查已实现的软件是否满足了需求规格说明中的各类需求以及软件配置是否完全、正确，验证系统的功能是否达到设计要求。

5.5.3 系统测试结果

按照用户角色划分进行全面的系统测试工作。整个测试过程中，共使用测试用例 98 个，测试结果正确的有 92 个，测试结果不完全正确的有 6 个，测试结果有较大错误的有 0 个。具体测试结果如表 5-15 所示：

表 5-13 系统测试结果

	总测试用例	测试结果正确	测试结果不完全正确	测试结果有较大错误
数目	98	92	6	0
百分比	100%	93.8%	6.2%	0%

通过对代码的修改和调试，进行系统回归测试，就是重新进行测试以确认修改没有引入新错误或导致其他代码产生错误。在进行三轮修改和回归测试后系统的测试结果正确率为 100%。

5.5.4 性能测试

性能测试包括压力测试、负载测试和并发测试等多种方法。Apache Jmeters^[51]是一个纯 Java 桌面应用测试工具,用于压力测试和性能测量。本文用负载测试来考察系统的性能。负载测试从保持预定义的事物处理速度的前提下处理特定数量并发用户的角度验证系统性能。

测试目标

- 1、功能模块在一定负载下的反应时间是多少
- 2、在一定负载下系统性能的参数指标是多少
- 3、系统主要压力承受点在什么地方
- 4、系统是否达到了性能需求的目标

测试计划

下面主要对两个功能：主页的链接和施工组织设计编制列表的查询分别进行单一的测试，分别执行以 25 递进分别从 50 到 550 个并发用户的情况下系统的反应情况。根据项目的具体设计，在 100 个用户时，90%的事务平均响应时间不超过 5 秒，每一事务的最大响应时间不超过 7 秒；在 300 个用户时，90%的事务平均响应时间不超过 8 秒，每一事务的最大响应时间不超过 10s。

测试过程

1、增加负载信息设置：设置线程数(Number of Threads)：50 到 550；Ramp-up Period:0;循环次数(Loop Count)：1

2、增加 Http 请求：

服务器名称或 Ip 地址：localhost

端口号：7001

路径：/ZhscWar/com/shchgl/shggl/sgzzsj/design/SelectAction.do

和 http://localhost:7001/ZhglWar/login.jsp

方法：get

3、增加监听器

添加聚合报告：总的样本数、最大最小平均响应时间

用表格查看结果：用表格显示线程测试情况

察看结果树：显示取样器结果请求和相应数据

测试说明

Samples:本次场景一共完成了多少个线程

Average:平均响应时间

Median:统计意义上响应时间的中值

90%Line:所有线程中 90%的线程响应时间都小于的值

Min:最小响应时间

Max:最大响应时间

Error:出错率

Throughput:吞吐量

KB/sec:以流量作为衡量的吞吐量

5.5.5 性能测试结果

(1)不同规模虚拟用户下的返回错误页面率分析

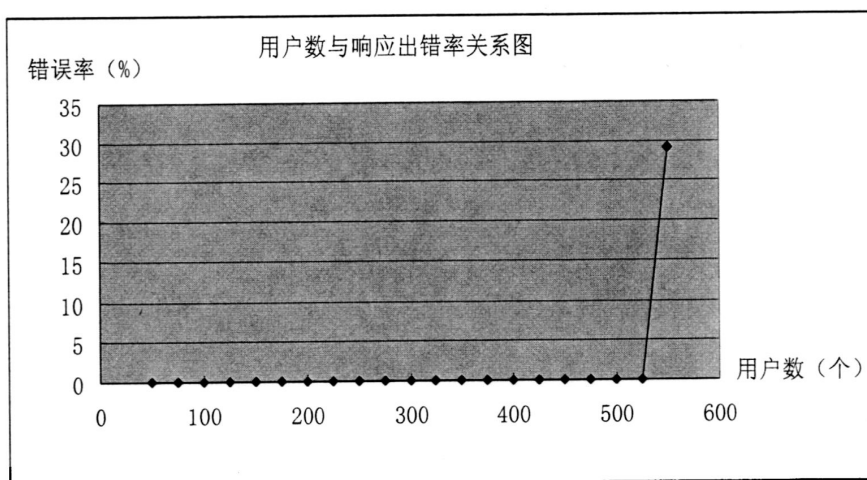


图 5-5 用户数与响应出错率关系图

从图 5-5 可以看出当用户数在 550 以下的时候服务器能正常响应全部用户的客户端请求。而一旦并发用户量达到 550 时服务器开始出现错误，并且出现响应超时现象。也就是说服务器在并发用户数达到 550 时已经不能正常工作，需要更新服务器硬件或者对系统进行优化。

(2)不同规模虚拟用户下的系统延迟分析:

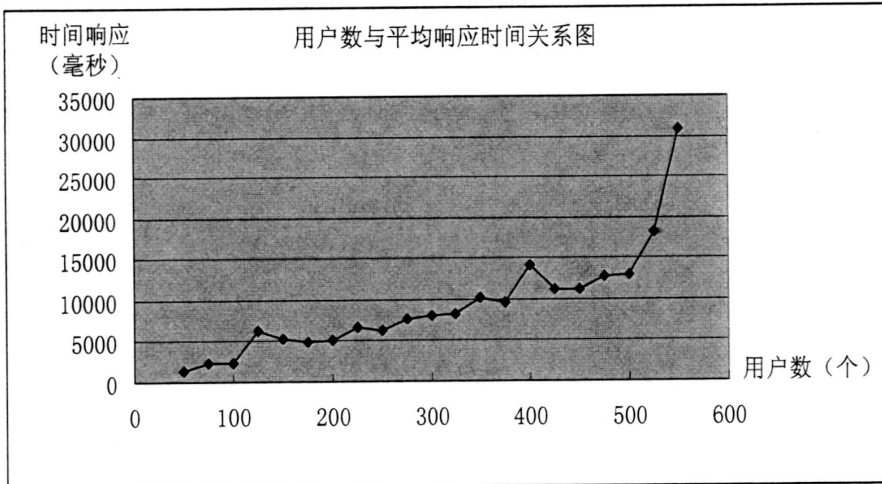


图 5-6 用户数与平均响应时间关系图

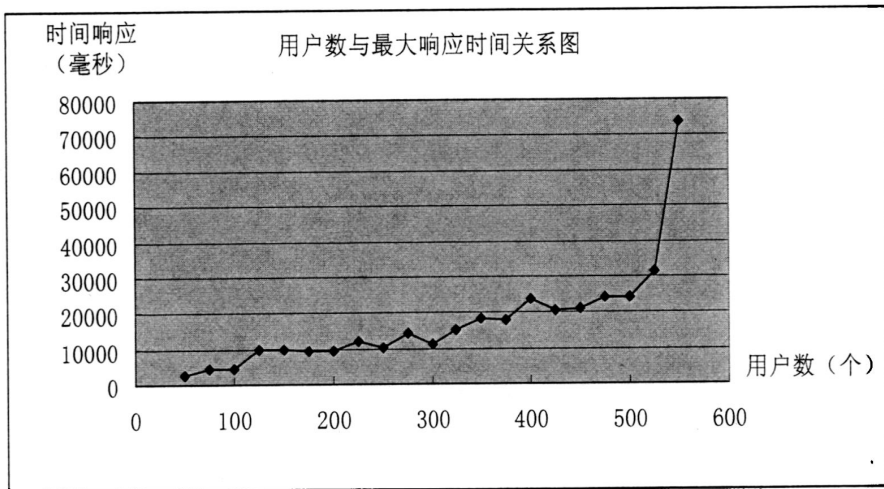


图 5-7 用户数与最大响应时间关系图

从图 5-6 和图 5-7 可以看出，在并发用户数为 100 的时候，延长最长的页面有 4.5s，而平均延长在 2.4s，说明绝大多数用户的行为在 3s 以内就能得到响应，用户的感受是相当不错的，一般认为，页面响应控制在 5s 以内用户的感受是令人满意的。当并发用户数达到 200 时，最大延迟 9.4s，平均延迟 5.1s，也就是说大多数网页的反应速度在 10s 以内，故用户体验良好。当并发用户数达到 350 时，在执行通过查询获得施工组织设计编制列表时反应时间超过 10s，用户体验很差，但服务器能正常工作。据此可以推断在理想状况下应该将同时在线的用户稳定在 300 以下，以保证良好的用户体验；而且并发用户的峰值也应该控制在 350 以下，从而保证系统的正常运行。

(3) 不同类型操作的反应时间分析

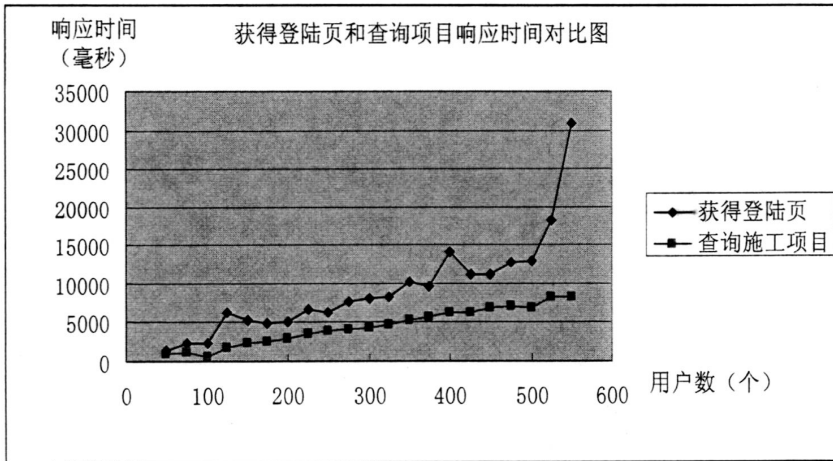


图 5-8 登陆主页与查询项目列表响应时间对比图

从 50 个用户到 550 个用户的测试过程中，无论在多少个并发用户数规模下，都可以看到 `SelectAction.do` 的相应时间总是大于 `login.jsp`，且在相同并发用户数下的错误率也比较高。从查看结果树可以看出，报错的代码是 `Http Response code:500, Internal Server error`。此时还无法判断出究竟是服务器硬件还是软件问题。

(4)不同虚拟用户数与吞吐量、服务器接收量关系分析

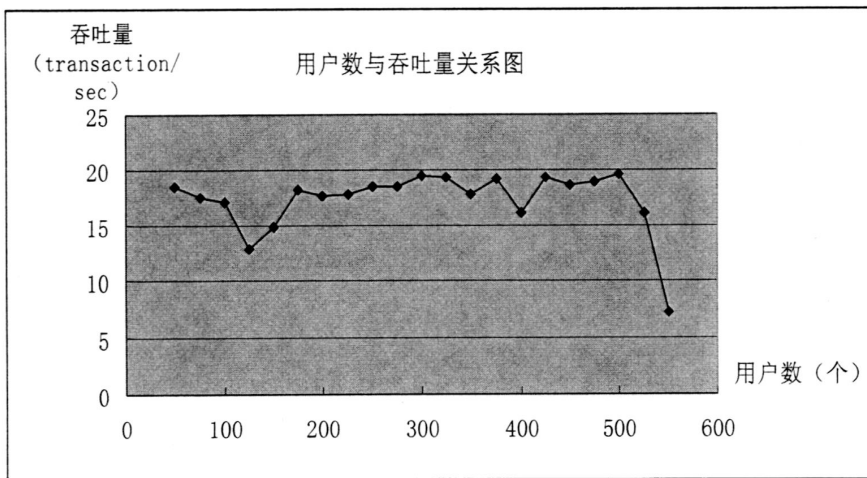


图 5-9 用户数与吞吐量关系图

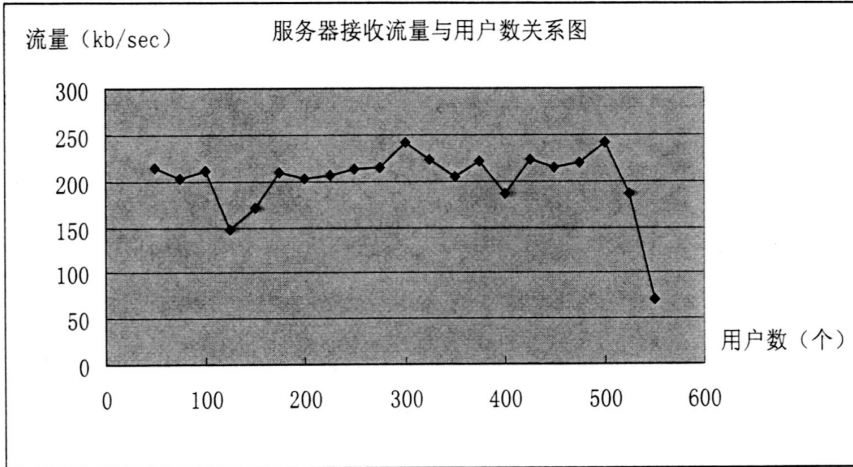


图 5-10 用户数与服务器接收流量关系图

最佳或理想状态下，服务器的吞吐量以及接受到的数据流量应该是随着用户数的增加而上升。图 5-9 和图 5-10 表明当用户数增加的同时服务器接受的吞吐量和请求数据流量并未上升，在接近 550 附近反而迅速下降。说明此时硬件性能资源出现瓶颈。

(5)虚拟用户数为 550 时的系统硬件性能分析

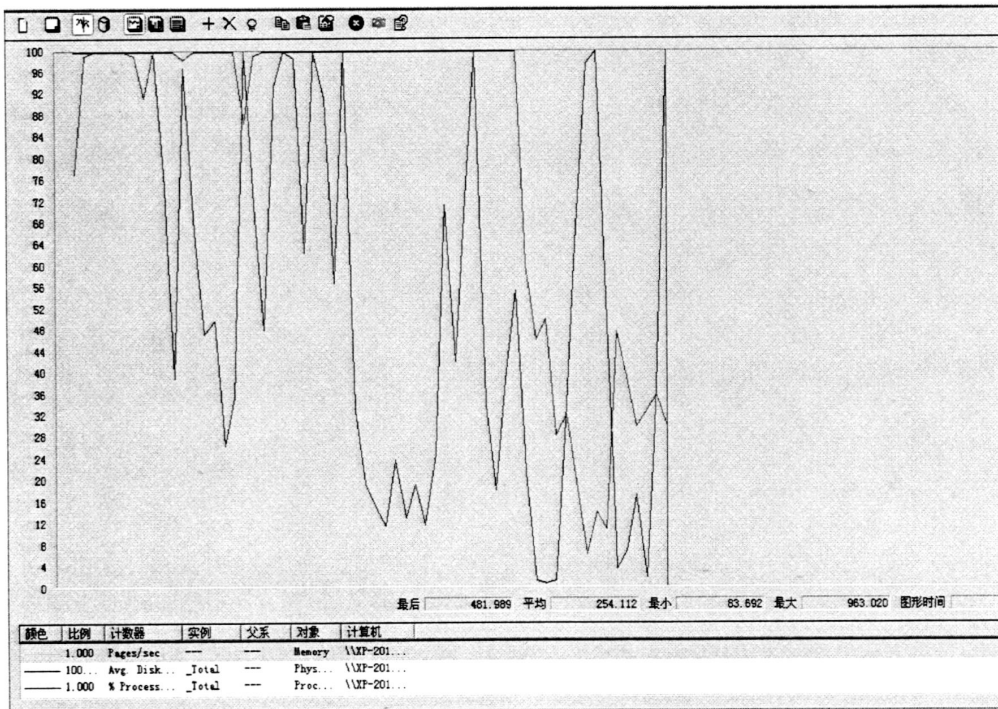


图 5-11 系统硬件性能监视图

图中黄线为 Pages/sec (内存分页/秒)，一直处于速率非常高的状态，说明此时调用内存分页十分频繁，系统的性能此时很差。

蓝线为 Avg.Disk Queue Length (磁盘访问序列的平均值)。

绿线为%Processor Time (处理器执行有效线程时间的百分比), 这个值在测试过程中多数时间保持在 90%以上, 说明此时 CPU 是系统瓶颈。

由图上还可以看出当磁盘访问序列值处于上升状态的时候, 页面读取率并没有降低, 说明此时内存是系统瓶颈。

(6)系统优化方案

测试是在局域网中完成的, 测试中使用的带宽远小于现有提供的带宽。排除网络带宽成为瓶颈的可能, 而测试中 CPU 使用率一直处于 90%以上, 且内存读取率也非常高, 所以应该对服务器的 CPU 和内存进行优化升级。

总结和展望

本文对客运专线基础设施维修基地维修管理系统施工管理子系统做了详细的分析设计，并基于 Java EE 架构，在 Beehive 框架基础上进行分层设计，完成了施工管理子系统的实现。通过 Jbpm workflow 引擎实现了针对施工计划的定制 workflow 系统。另外论文还提出一种既满足铁路客运专线业务又能灵活扩展的通用权限平台设计，提高了系统的安全性。系统在运行过程中各方面都达到预期目标，系统的开发和论文撰写的过程，也是自己对 Beehive 框架、Jbpm workflow 引擎、权限控制等更加深入理解的过程。主要工作总结如下：

(1)在 Beehive 框架的基础上利用 controls 控件技术对业务逻辑中数据库业务操作增、删等进行封装形成基础服务层。在开发中业务逻辑层几乎不涉及 SQL 语句编写，这样对于提高开发速度起到显著的作用。

(2)充分调研客运专线维修基地施工作业流程，对项目中施工组织设计、年度轮廓计划、月度计划、日计划等计划的编制，施工过程中的进度，施工完成后的检验验收进行监控和管理。在进行科学的需求分析，并建模，最后完成各功能的实现。

(3)设计基于 JBPM 施工日计划的申报流程并实现上报审批功能采用开源 workflow 引擎 JBPM 设计施工日计划的上报审批流程，实现业务和流程相分离，提供自定义流程模板导入，对整个流程的状态和节点实时状态监控。

(4)提出一种既满足铁路客运专线业务又能灵活扩展的通用权限平台。对基于角色的访问控制 RBAC 模型进行改进，设计符合我国铁路信息化建设需求的 RUP 模型。

下一步要完成的工作：

(1)将施工日计划、施工月计划和既有线的运行调度系统对接，这样施工类计划在客专维修基地各级上级部门的审批部分就可以在运行调度系统中完成，这样更有利于上级相关部门对施工计划的审批，提高审批效率。

(2)在施工计划流程上报中增加系统消息提醒或邮件提醒的功能。

致 谢

首先衷心感谢我的导师王小敏老师，王老师广博的学识、严谨勤奋的科研态度、谦和的待人哲学让我终生难忘，他是我学习的榜样。

在研究生学习期间，王老师为我创造了良好的学习与研究平台。不仅在学习上和项目上给予指导，在思想上也给我极大启发。教我做人做事的道理。我所取得的每一点进步都倾注了他的心血，再次表示我诚挚的感谢。

感谢西南交通大学信息科学与技术学院诸位老师三年来的教导，感谢你们对我的栽培和教育。

感谢 1504 实验室的各位同学，师弟师妹，和你们融洽相处的时光，值得我一生回忆。

感谢家人一直以来的支持、关心和理解，是你们推动着我前进。

最后，我还要衷心感谢为评阅本论文而付出辛勤劳动的各位评审老师。感谢你们对我论文的赐教和指正。

参考文献

- [1] 杨章韬.客运专线对区域经济的影响[D].中南大学硕士学位论文. 2010.
- [2] 李倩.客运专线动态可视工程施工管理信息系统关键技术研究[D],中南大学博士学位论文.2010.
- [3] Derek Miers.BPM Driving Business Performance[EB/OL]. <http://www.bpmfocus.org>, 2008.4.
- [4] BEA Systems. State of the BPM Market[EB/OL].
- [5] 上海东兰科技发展有限公司.LiveFlow 工作流程管理系统 [EB/OL]. <http://www.dltech.com.cn/liveflow/liveflow.htm>, 2008.5.
- [6] 信雅达系统工程股份有限公司.信雅达工作流程管理系统 [EB/OL] <http://ecology.sunyard.net.cn/weaver/weaver.file.FileDownload?fileid=436,2008.6>
- [7] 神州数码管理系统有限公司.Easyflow 工作流程管理系统 [EB/OL] http://www.dreamer.com.cn/product_and_server/easyflow.htm, 2008.5
- [8] 上海携创软件技术有限公司. Joinwork 业务流程管理软件 [EB/OL].
- [9] JBoss Corporation. JPD L User Guide:Introduction of JBPM[EB/OL]. <http://docs.jboss.com/jbpm/v3/userguide/>,2008.6
- [10] 李兴华. Java Web 开发实战经典. 北京:清华大学出版社[M], 2010
- [11] 廖雄飞.Beehive 面向 J2EE 和 SOA 的应用框架.现代计算 [J],2005, 223:76-77.
- [12] http://dev.yesky.com/403/2582903_1.shtml,2006.9.
- [13] Workflow Management Coalition.Terminology and Glossary.www.huihoo.org/jfox/jfoxflow/specification/03.Terminology_Glossary.pdf,2008.7.
- [14] 谢艳平. 基于 J2EE 和 Jbpm 的分布式工作流的研究与应用[D]. 武汉理工大学硕士学位论文, 2006.
- [15] Workflow Management Coalition.Terminology and Glossary.www.huihoo.org/jfox/jfoxflow/specification/03.Terminology_Glossary.pdf,2008.7
- [16] Workflow Management Coalition.Process Definition Interchange Organization Model. www.huihoo.org/jfox/jfoxflow/specification/04.interface1_The_Process_Definition_Interchange_Process_Model.pdf,2008.7
- [17] 症代峰.基于 JBPM 的业务流程系统的研究与实现[D]. 武汉理工大学硕士学位论文, 2009.
- [18] 侯瑞春, 高健, 刘学义.JAVA 设计模式在工作流管理系统中的应用研究

- [J]. 微型计算机. 2010, 26:176-178.
- [19] 高杰. 深入浅出JBPM. 人民邮电出版社[M],2009.
- [20] 汪海涛.JBoss JBPM workflow技术的改进及应用研究[D]. 武汉理工大学硕士学位论文,2008.
- [21] JBoss Corporation. Javadoc-jpdl[EB/OL]
- [22] 谭伟, 范玉顺.业务过程管理框架与关键技术研究[J].计算机集成制造系统, 2004,10(7):737-745.
- [23] 王强, 罗四维.商业流程执行语言的研究与应用[J].铁路计算机应用, 2004, 13(5):26-28.
- [24] Crawford W.Kaplan.J2EE Design Patterns.US:WroxPress,2003
- [25] 阎宏. Java与模式.电子工业出版社[M], 2007.
- [26] 李刚. Struts2权威指南--基于Webwork核心的MVC开发.电子工业出版社, 2007.9.1.
- [27] Allamarajus, Bust C.Davies Jetal.Professional Java Server Programming J2ee 1.3 Edition.
- [28] MartyHall. Servlet与JSP权威指南.北京:机械工业出版社, 2002.
- [29] 于海霞, 基于J2EE的WEB应用框架的研究与应用[D], 武汉理工大学硕士学位论文, 2007.
- [30] 赵强.基于开源软件的J2EE企业级应用开发.北京:电子工业出版社, 2005
- [31] 刘洋, 魏飞, 精通JBOSS-EJB与WebService开发精解.电子工业出版社, 2004.
- [32] 张兵. 基于轻量级J2EE的电力物资管理系统的设计与实现. 西南交通大学硕士学位论文[D], 2008.
- [33] James J.Dell.Advanced Object-Oriented Analysis&Design Using UML. Cambridge University Press,1988.
- [34] 邵维忠等著.统一建模语言UML述评[J].计算机研究与发展. 1999,36(4):385-394.
- [35] Khawar Zaman Ahmed.UML与J2EE企业级应用开发.科学出版社. 2004 :23-35.
- [36] 萨师煊, 王珊. 数据库系统概论.高等教育出版社.1983.1:80-82.
- [37] Thomas M.Connly Carolyn E.Begg.数据库设计教程.机械工业出版社, 2005
- [38] 陶宏才 数据库原理及设计. 清华大学出版社[M], 2007.
- [39] 肖军模, 刘军, 周海刚.网络信息安全.机械工业出版社, 2006:37-90.
- [40] 崔宾阁, 刘大昕.强制访问控制在基于角色的保护系统中的实现.计算机工程[J], 2006, 32(6):167-169.

-
- [41] 魏立峰,孟凯凯,何联跃. 面向用户角色的细粒度自主访问控制机制[J]. 计算机应用, 2009,10.
- [42] Dong GY, Qing SH, Liu KL, Role-Based authorization constraint with time character[J]. Journal of Software, 2002,13(8):1521-1527.
- [43] Ravi Sandhu, Edward J.Coyne, Hal L.Feinstein,Charles E.Youman.Role-Based Access Control Models[J]. IEEE Computer,1996,29(2):38-47.
- [44] R.Sandhu,V.Bhamidipati,Q.Munawer.The ARBAC97Model for Role-Based Administration. of Roles[J].ACM Transaction on Information and System Security,1999,2(1):105-135
- [45] 毕可骏. 基于SSH架构的多层分级党务管理信息系统的设计与实现. 西南交通大学硕士学位论文[D], 2009.
- [46] <http://www.cnblogs.com/thinhunan/archive/2006/04/01/DeveloperNotesForPrototype.html>, 2006.4
- [47] <http://www.junit.org/>.
- [48] 史明慧. 基于组合框架的J2EE组件单元测试的研究与应用[D]. 中国农业大学硕士论文.2005:32-33.
- [49] <http://httpunit.sourceforge.net/>
- [50] 李红岩. 一种J2EE平台下整合开源框架测试技术的研究与实现[D]. 燕山大学硕士论文, 2006
- [51] 王宇剑.Jmeter在消息中间件系统压力测试中的应用[D]. 浙江大学硕士学位论文, 2006:35-45
-

攻读硕士学位期间发表的论文

- [1] 胡同宇. 客运专线基础设施维修基地管理信息系统设计. IECT, 2012