****

**本 科 毕 业 设 计**

**院 系** 软件学院

**专 业** 软件工程

**题 目** 智造链系统的设计部模块、生产部模块和

质检部模块的设计与实现

**年 级** 2010级 **学 号** 101250126

**学生姓名** 孙克维

**指导教师** 葛季栋 **职 称** 讲师

**论文提交日期** 2014年5月25日

**南京大学本科生毕业论文（设计）中文摘要**

毕业论文题目：智造链系统的设计部模块、生产部模块和质检部模块的设计与实现

软件学院 院系 软件工程 专业 2010 级本科生姓名： 孙克维

指导教师（姓名、职称）： 葛季栋 讲师

摘要：

在传统的订单管理流程中，一个订单的整个管理过程往往需要很多角色的参与，如市场部、生产部、采购部等部门的员工，而这些角色之间沟通交流的方式多种多样，会增加流程的复杂度以及订单管理的困难程度，同时流程当前的状态，即流程进行到哪一步也难以跟踪。而一个使用工作流框架的系统则能较好的解决这些问题。

本文介绍的智造链系统将不同角色的操作分割成单独的模块，为每个角色提供不同的交互页面，使每个角色的任务都能被简洁明了的呈现出来，原本复杂的订单管理流程变得简单，大大提高了订单管理的效率。

智造链系统采用B/S架构，使用jBPM、Spring、Hibernate等框架和技术进行开发。jBPM是一种基于JavaEE的轻量级工作流管理系统，用java语言编写。JBPM的流程建模模型是基于Activity Diagram（活动图）的，并在引擎构建上融入了FSM和PetriNet思想，允许程序员在其整个生命周期中进行建模，并执行和监控业务流程。

智造链系统包括多个模块，分别是市场部模块、采购部模块、设计部模块、生产部模块、采购部模块、物流部模块、质检部模块、系统管理员模块以及通信模块。

本人在项目开发过程中负责设计部模块、生产部模块以及质检部模块。

关键词：智造链系统、工作流技术、jBPM、设计部模块、生产部模块、质检部模块

**南京大学本科生毕业论文（设计）英文摘要**

THESIS: Design and implementation of design module, produce module and quality module in zhizaolian System

DEPARTMENT: Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: 2010

MENTOR: Jidong Ge

ABSTRACT:

In the traditional order management process, an entire order management process often requires the participation of many roles, such as marketing department, production department, purchasing department and other departments , and the communication between these roles will increase the complexity and difficulty of the order management process, and the current state of the process is difficult to trace as well. The use of a workflow system framework is able to solve these problems.

Zhizaolian system described in this article will split operations of different roles into separate modules to provide different interactive pages for each role, so that the task of each role can be simply presented. The complex order management process becomes easier, greatly improving the efficiency of order management.

Zhizaolian system uses B/S structure, using jBPM, Spring, Hibernate frameworks and technology for development. jBPM is a lightweight JavaEE -based workflow management system, written in java language. jBPM process modeling model is based on Activity Diagram, and has the thinking of FSM and PetriNet in the construction of the engine, allowing programmers to model throughout its lifecycle , and to execute and monitor business processes.

Zhizaolian system includes a plurality of modules, such as marketing module, purchasing module, design module and so on. The author is responsible for design module, production module and quality module in the project development process.

KEY WORDS: zhizaolian System, workflow technology, jBPM, design module, production module, quality module

目录

[图目录 IV](#_Toc387496427)

[表目录 V](#_Toc387496428)

[第一章 引言 1](#_Toc387496429)

[1.1 项目背景 1](#_Toc387496430)

[1.2 工作流技术的发展历史 1](#_Toc387496431)

[1.3 论文的主要工作和组织结构 2](#_Toc387496432)

[第二章 智造链系统技术概述 3](#_Toc387496433)

[2.1 jBPM 3](#_Toc387496434)

[2.1.1 jBPM5结构 3](#_Toc387496435)

[2.1.2 jBPM5的过程定义语言分析 4](#_Toc387496436)

[2.2 Spring框架 5](#_Toc387496437)

[2.2.1 Spring简介 5](#_Toc387496438)

[2.2.2 Spring结构 5](#_Toc387496439)

[2.3 Hibernate 7](#_Toc387496440)

[2.3.1 Hibernate简介 7](#_Toc387496441)

[2.3.2 Hibernate优势特点 7](#_Toc387496442)

[2.3.3 Hibernate结构 8](#_Toc387496443)

[2.4 本章小结 8](#_Toc387496444)

[第三章 智造链系统需求分析与概要设计 9](#_Toc387496445)

[3.1 智造链系统整体概述 9](#_Toc387496446)

[3.1.1 系统用户角色 9](#_Toc387496447)

[3.1.2 系统功能性需求 9](#_Toc387496448)

[3.1.3 系统非功能性需求 12](#_Toc387496449)

[3.1.4 系统运行环境 12](#_Toc387496450)

[3.2 智造链系统设计部模块的需求分析 12](#_Toc387496451)

[3.2.1 设计部模块用例描述 12](#_Toc387496452)

[3.2.2 设计部模块用例图 13](#_Toc387496453)

[3.2.3 设计部模块实体关系分析 14](#_Toc387496454)

[3.3 智造链系统生产部模块的需求分析 15](#_Toc387496455)

[3.3.1 生产部模块用例描述 15](#_Toc387496456)

[3.3.2 生产部模块用例图 16](#_Toc387496457)

[3.3.3 生产部模块实体关系分析 17](#_Toc387496458)

[3.4 智造链系统质检部模块的需求分析 18](#_Toc387496459)

[3.4.1 质检部模块用例描述 18](#_Toc387496460)

[3.4.2 质检部模块用例图 19](#_Toc387496461)

[3.4.3 质检部模块实体关系分析 19](#_Toc387496462)

[3.5 智造链系统设计部模块的概要设计 20](#_Toc387496463)

[3.5.1 设计部模块程序结构设计 20](#_Toc387496464)

[3.5.2 设计部模块功能组件划分 21](#_Toc387496465)

[3.5.3 设计部模块接口设计 21](#_Toc387496466)

[3.6 智造链系统生产部模块的概要设计 22](#_Toc387496467)

[3.6.1 生产部模块程序结构设计 22](#_Toc387496468)

[3.6.2 生产部模块功能组件划分 23](#_Toc387496469)

[3.6.3 生产部模块接口设计 23](#_Toc387496470)

[3.7 智造链系统质检部模块的概要设计 24](#_Toc387496471)

[3.7.1 质检部模块程序结构设计 24](#_Toc387496472)

[3.7.2 质检部模块功能组件划分 24](#_Toc387496473)

[3.7.3 质检部模块接口设计 24](#_Toc387496474)

[3.8本章小结 25](#_Toc387496475)

[第四章 智造链系统设计部模块、生产部模块与质检部模块的详细设计与实现 26](#_Toc387496476)

[4.1 设计部模块、生产部模块和质检部模块概述 26](#_Toc387496477)

[4.1.1 设计部模块概述 26](#_Toc387496478)

[4.1.2 生产部模块概述 26](#_Toc387496479)

[4.1.3 质检部模块概述 27](#_Toc387496480)

[4.2 设计部模块的详细设计 27](#_Toc387496481)

[4.2.1 设计部模块详细类图 27](#_Toc387496482)

[4.2.2 设计部模块顺序图 29](#_Toc387496483)

[4.2.3 设计部模块关键数据表设计 30](#_Toc387496484)

[4.3 生产部模块的详细设计 30](#_Toc387496485)

[4.3.1 生产部模块详细类图 30](#_Toc387496486)

[4.3.2 生产部模块顺序图 32](#_Toc387496487)

[4.3.3 生产部模块关键数据表设计 33](#_Toc387496488)

[4.4 质检部模块的详细设计 33](#_Toc387496489)

[4.4.1 质检部模块详细类图 33](#_Toc387496490)

[4.4.2 质检部模块顺序图 35](#_Toc387496491)

[4.4.3 质检部模块关键数据表设计 36](#_Toc387496492)

[4.5 设计部模块的实现 36](#_Toc387496493)

[4.5.1 设计验证 36](#_Toc387496494)

[4.5.2 录入样衣版型 38](#_Toc387496495)

[4.5.3 设计生产验证 39](#_Toc387496496)

[4.5.4 生产版型确认 39](#_Toc387496497)

[4.6 生产部模块的实现 39](#_Toc387496498)

[4.6.1 生产验证 39](#_Toc387496499)

[4.6.2 生产成本核算 40](#_Toc387496500)

[4.6.3 样衣生产 41](#_Toc387496501)

[4.6.4 批量生产 41](#_Toc387496502)

[4.7 质检部模块的实现 43](#_Toc387496503)

[4.8本章小结 43](#_Toc387496504)

[第五章 总结与展望 44](#_Toc387496505)

[5.1 总结 44](#_Toc387496506)

[5.2 展望 44](#_Toc387496507)

[参考文献 45](#_Toc387496508)

[致谢 46](#_Toc387496509)

# 图目录

[图2.1 Eclipse图形化插件 5](#_Toc389557395)

[图2.2 基于Spring框架的完整Web应用体系结构 6](#_Toc389557396)

[图2.3 Hibernate结构图 8](#_Toc389557397)

[图3.1 智造链系统模块图 11](#_Toc389557398)

[图3.2 设计部模块用例图 14](#_Toc389557399)

[图3.3 设计部模块实体关系图 15](#_Toc389557400)

[图3.4 生产部模块用例图 17](#_Toc389557401)

[图3.5 生产部模块实体关系图 18](#_Toc389557402)

[图3.6 质检部模块用例图 19](#_Toc389557403)

[图3.7 系统程序结构图 20](#_Toc389557404)

[图3.8 系统功能组件划分图 21](#_Toc389557405)

[图4.1 设计部模块详细类图 28](#_Toc389557406)

[图4.2 设计部模块顺序图 29](#_Toc389557407)

[图4.3 生产部模块详细类图 31](#_Toc389557408)

[图4.4 生产部模块顺序图 32](#_Toc389557409)

[图4.5 质检部模块详细类图 34](#_Toc389557410)

[图4.6 质检部模块顺序图 35](#_Toc389557411)

[图4.7 设计部模块设计验证任务列表页面 36](#_Toc389557412)

[图4.8 设计部模块设计验证任务详情页面 37](#_Toc389557413)

[图4.9 DesignService类verifyDesignSubmit方法代码 37](#_Toc389557414)

[图4.10 设计部模块录入样衣版型页面 38](#_Toc389557415)

[图4.11 DesignService类uploadDesignSubmit方法代码 39](#_Toc389557416)

[图4.12 生产部模块生产成本核算页面 40](#_Toc389557417)

[图4.13 ProduceService类computeProduceCostSubmit方法代码 41](#_Toc389557418)

[图4.14 生产部模块批量生产页面 41](#_Toc389557419)

[图4.15 ProduceService类pruduceSubmit方法代码 42](#_Toc389557420)

# 表目录

[表3.1 DesignService接口方法表 21](#_Toc387496614)

[表3.2 DesignCadDAO接口方法表 22](#_Toc387496615)

[表3.3 ProduceService接口方法表 23](#_Toc387496616)

[表3.4 IProduceDAO接口方法表 23](#_Toc387496617)

[表3.5 QualityService接口方法表 24](#_Toc387496618)

[表4.1 cad文件信息表 30](#_Toc387496619)

[表4.2 加工单信息表 33](#_Toc387496620)

# 第一章 引言

## 1.1 项目背景

传统的服装生产一般都是一个订单会生产大量的产品，而智造链公司则是专门针对那些生产数量不大的订单，能够较为快速的完成客户的需求。在传统的服装生产行业中，从客户下订单到生产完成，往往要经历很多繁琐的步骤，而这些步骤一般由员工手工记录完成，一些信息如客户信息、订单信息等，一般内容很多，容易造成查找的不便以及遗失等问题。相对于传统的服装生产厂商，智造链这样的公司则会有更多的订单量，这会产生更加严重的信息管理方面的问题，这时，能够更加快速有效的管理订单以及客户等信息就显得尤为重要。传统的以笔和纸或者普通的文档为媒介的管理方式显然已经不足以胜任如此繁重的任务。

在传统的订单管理流程中，一个订单的整个管理过程往往需要很多角色的参与，如市场部、生产部、采购部等部门的员工，而这些角色之间沟通交流的方式多种多样，会增加流程的复杂度以及订单管理的困难程度，同时流程当前的状态，即流程进行到哪一步也难以跟踪。而一个使用工作流框架的系统则能较好的解决这些问题。

工作流通过将整个工作流程按角色分解成较小的任务，按照一定的规则和流程来执行这些任务，并对这些过程进行监控，达到增强管理和提高效率的目的[1][2]。本文介绍的智造链系统使用jBPM工作流框架，将整个订单管理流程系统的结合在一起，统一管理，方便各参与角色的互动交流，提高订单管理的效率，解决了以上提到的诸多问题。

## 1.2 工作流技术的发展历史

工作流技术提出的目的是通过将工作分解成定义良好的任务、角色，按照一定的规则和过程来执行这些任务并对他们进行监控，从而达到提高工作效率、降低生产成本、提高企业生产经营管理水平和企业竞争力的目标。[3]工作流思想的出现可以向上追溯到1968年，Fritz Nordsieck就已经清楚地表达了利用信息技术实现工作流程自动化的想法。而工作流技术起源于20世纪70年代中期办公自动化领域的研究，其中SCOOP和Officetalk系统不但标志着工作流技术的开始，而且也是最早的办公自动化系统。但由于个人计算机使用以及网络技术在当时尚未普及且相关技术在理论基础方面的匮乏，导致这项新技术并未取得成功。

20世纪80年代，FileNet和ViewStar等公司作为最早一批工作流产品供应商，他们对外提供一种全过程对某些业务流程进行支持的集成化软件，即早期出现的工作流管理系统。这些系统主要来自于图像处理领域和电子邮件领域，其中这些早期的工作流系统中只有少数获得了较好发展。

到了20世纪90年代，个人计算机、网络技术日新月异，信息化建设日臻完善，在实现大规模的异构分布执行环境这一必然趋势的指引下，工作流系统由起初的创建无纸化办公环境，逐步向同化企业复杂信息环境、实现业务流程自动执行这一方向转变，工作流技术的研究与开发进入了一个新的热潮。[4]

工作流技术标准化的工业组织——工作流管理联盟（WFMC）成立于1993年8月，它的成立标志着工作流技术从计算机应用研究领域中脱离出来，成为一门独立的技术学科，工作流技术的研究进入到一个相对成熟的阶段。到1994年，该组织制定了一系列的工业标准并发布了有关工作流管理系统的工作流参考模型。[5]当下在全球范围内，对工作流的技术研究以及相关的产品开发进入了更为繁荣的阶段，更多更新的技术被集成进来，文件管理系统、数据库、电子邮件、移动式计算、Internet服务等都已被容纳到工作流管理系统中。[6]因而利用工作流方法进行业务过程建模和深入分析，优化和重组业务流程中的不合理环节这一项技术具有十分重大的意义。

## 1.3 论文的主要工作和组织结构

本文介绍了智造链系统的设计部模块、生产部模块以及质检部模块的设计与开发细节。

第一章，引言部分，主要介绍了智造链系统的项目背景，以及工作流技术的发展历史，并描述了论文的主要工作和组织结构。

第二章，智造链系统的技术概述，主要为项目中使用的相关技术，包括jBPM工作流框架、Spring框架、Hibernate等。

第三章，描述了智造链系统的功能系统和非功能性需求，介绍了设计部模块、生产部模块和质检部模块的需求分析和概要设计，需求分析包括用例描述、用例图和实体关系图等，概要设计包括程序结构设计和接口设计等。

第四章，描述了智造链系统设计部模块、生产部模块和质检部模块的详细设计和实现细节，详细设计包括详细类图、顺序图和数据表设计等，实现细节包括页面截图和部分功能代码。

第五章，总结与展望，对整个项目进行总结，讨论该系统的缺点与不足，并对该系统的后续版本和前景做展望。

# 第二章 智造链系统技术概述

## 2.1 jBPM

jBPM全称是Java Business Process Management（业务流程管理），是一种基于JavaEE的轻量级工作流管理系统，用java语言编写。JBPM的流程建模模型是基于Activity Diagram（活动图）的，并在引擎构建上融入了FSM和PetriNet思想，允许程序员在其整个生命周期中进行建模，并执行和监控业务流程。

由于jBPM架构的开放性，它更像是一个支持面向流程编程的框架，覆盖了工作流、服务协作、业务流程管理等领域的一个开源的灵活的、易扩展的可执行流程语言框架。jBPM是公开源代码项目，它的使用要遵循LGPL（GNU Lesser General Public License）和EULA（JBoss End User License Agreement）协议。[7]

jBPM采用直观的流程定义语言jPdl(jBPM Process Definition Language)。jPdl认为一个业务流程可以被看作是一个UML状态图，详细定义了这个状态图的每个部分，如初始、结束状态以及状态之间的转换等等。jBPM还具有另一个鲜明的特色，即使用Hibernate来管理其数据库。Hibernate是目前Java领域最好的数据持久层解决方案。通过Hibernate，jBPM将数据的管理职能分离出去，自己专注于业务逻辑的处理。[8]

jBPM只有最小的依赖，可以像适用Java库一样十分容易的适用它。除此之外，也可以将其部署在J2EE集群应用服务器中，用于吞吐量极为关键的环境之中。同时jBPM可被配置为任何数据库，并能够部署到任何应用服务器。

### 2.1.1 jBPM5结构

jBPM5核心引擎（Core Engine）是整个项目中最为重要的部分。它是用来执行业务的一个轻量级工作流引擎。核心引擎可以作为应用的一个部分嵌入系统，当然也可部署作为服务。

以下列举核心引擎最重要的几个特征：

* 在对业务流程进行建模和执行时使用最新版本的BPMN2.0规范有着原生的支持；
* 着重强调其性能和可扩展性，执行流程实例时的可靠性和稳定性；
* 轻量级（可以嵌入到绝大多数支持简单Java运行环境的设备而不需要任何Web容器）

核心引擎同样可以集成一些其他的服务，如人工任务服务（Human Task Service）和历史日志（History Log）。人工任务服务是指当人类这一特定角色参与到流程中时用来管理人工任务。历史日志可以保存所有工作流引擎上执行的流程信息，可以用来保存当前或历史上所有已经完成的流程实例，也可以被用来查询流程实例的执行情况以及监测、分析相关的信息。

jBPM5使用一个或多个知识仓库（Repository）存储业务流程信息，基于web的设计工具被集成在Guvnor Repository上，知识仓库面向业务分析人员，可以不依赖与应用独立地管理业务流程。

### 2.1.2 jBPM5的过程定义语言分析

jBPM5不再直接支持JPDL过程定义语言的执行，而是使用BPMN2.0流程定义语言。2011年3月，BPMN2.0规范由OMG组织正式发布。

如图2.2所示，jBPM5提供了基于Eclipse的图形化插件，利用这个插件可以十分便捷地定义基于BPMN2.0规范的流程。基于图2.2所定义的流程，对BPMN2.0的一些基础元素进行简要的说明：

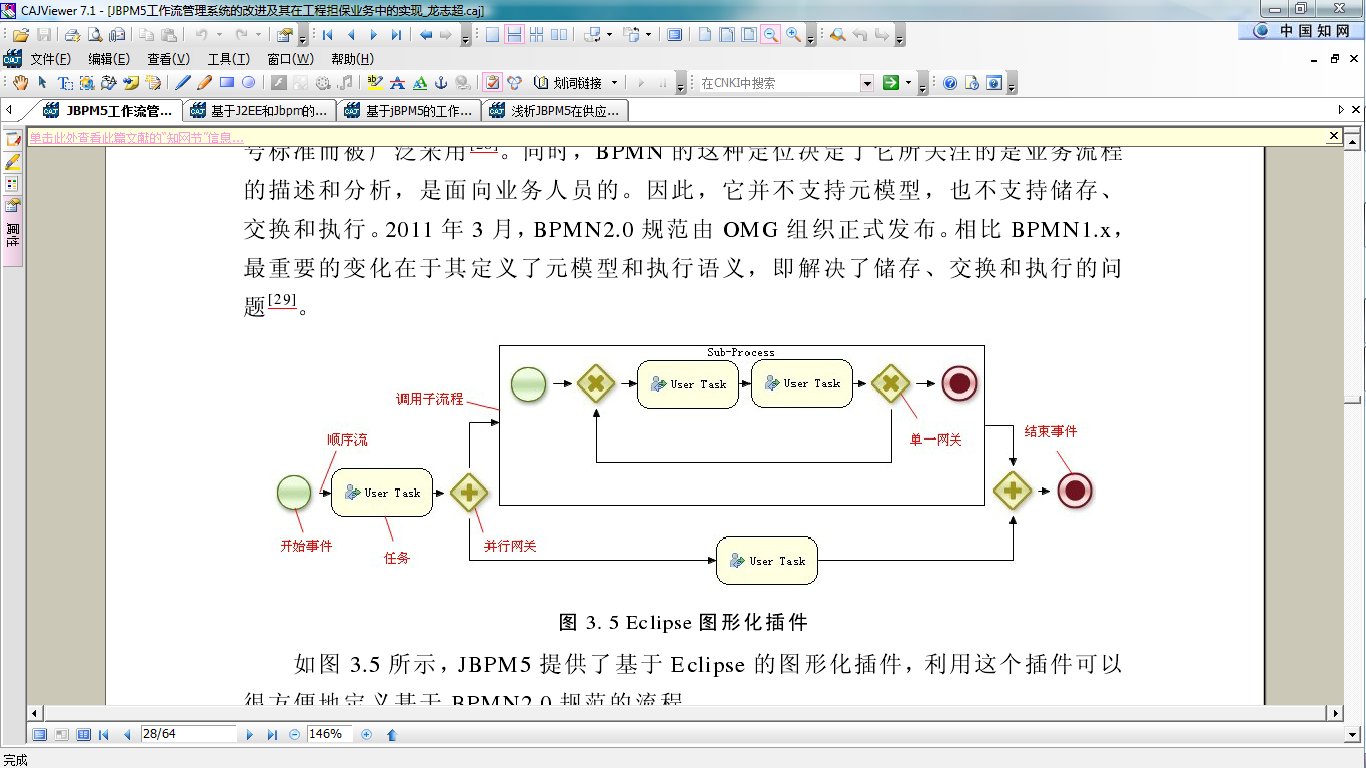
1. 开始事件（Start Event）：标志了一个流程将要开始。
2. 结束事件（End Event）：标志了一个流程将要结束。
3. 活动（Activity）：流程在流转过程中的一个一般工作项，任务和子流程是它的两种类型。
4. 任务（Task）：流程在流转过程中的一个原子活动，被用于在一个流程中的工作不能够被拆开到更为详细级别的情景。
5. 子流程(Sub-Process)：流程在流转过程中的一个复合活动，可以被进一步拆分为一组子活动。
6. 顺序流（Sequence Flow）：两个流程元素的连接器。一个元素在流程执行期间被访问后，流程会沿着该元素输出的顺序流继续执行。
7. 单一网关（Exclusive Gateway）：用于对流程中的决策进行建模。流程执行到这种网关时，按照输出流定义的顺序对他们进行计算。条件为真的顺序流被选取继续执行流程。
8. 并行网关（Parallel Gateway）：用来对流程中的并发进行建模，能够拆分出多个执行路径，或多个输入路径进行合并。其功能要根据输入和输出的顺序流进行划分。[4]

图2.1 Eclipse图形化插件

## 2.2 Spring框架

### 2.2.1 Spring简介

Spring Framework是由Rod Johnson创建的一个开放源码的应用框架。它是轻量级J2EE应用程式框架，旨在简化J2EE的开发，降低其项目实施难度。Spring Framework有利于为J2EE应用提供了基础结构，还为开发人员更好的适用J2EE相关技术提供了工具性支持。

Spring实现了控制反转技术（IoC），可以达到系统元件的松散耦合（Loose coupling），使得整个应用系统在架构上与维护上都能得到相当程度的简化。Spring的目标是实现一个全方位的整合框架，在Spring框架下实现多个子框架的组合，子框架之间彼此独立，也可以使用其它框架方案加以替代。Spring框架可以成为企业级应用程序一站式的解决方案，同时也是模块化的框架，允许开发人员使用其中一部分的模块进行开发。Spring框架是一个松耦合的框架，框架的部分耦合度被设计为最小，在各个层次上具体选用哪个框架取决于开发人员的需要。

### 2.2.2 Spring结构

Spring框架由7个模块组成，如图2.3所示。Spring模块构建在核心容器上，核心容器定义了创建、配置和管理bean的方式。在实际的设计中可以酌情的增加、减少或者加入其它的框架。

**Spring Core**

支持Bean容器管理的设施

**Spring Web**

**MVC**

提供Web MVC框架，可把JSP、velocity等技术实现的视图层装入框架

**Spring Context**

ApplicationContext提供校验、用户界面

**Spring Web**提供WebApplicationContext

**Spring AOP**

基于源代码级的元数据AOP设施

**Spring DAO**

提供事务处理器支持JDBC和DAO

**Spring ORM**

支持Hibernate等映射工作

图2.2 基于Spring框架的完整Web应用体系结构

* 核心容器（Spring core）：提供 Spring 框架的基本功能，其它模块均建立在此模块基础之上。核心容器主要组件是BeanFactory，它是工厂模式的实现。使用控制反转（IoC）模式将应用程序的配置和依赖性规范与实际的应用程序代码分开。
* 上下文模块（Spring Context）：构建于Bean包上Context包，提供了一种框架式的Bean访问方式对核心模块进行了扩展。
* AOP支持模块（Spring AOP）：通过配置管理特性，Spring AOP模块直接将面向方面的编程功能集成到了Spring框架中。所以，可以很容易地使Spring框架管理的任何对象支持AOP。Spring AOP模块为基于Spring的应用程序中的对象提供了事务管理服务。
* 数据存取对象模块（Spring DAO）：提供了有意义的异常层次结构，可用该结构来管理简化异常处理和不同数据库供应商抛出的错误消息。该包提供了一种方法实现编程性和声明事务管理。
* 对象关系映射模块（Spring ORM）：提供了对对象关系映射的支持，对流行的关系对象映射提供了集成层，其中包括 JDO、Hibernate 和 iBatis SQL Map。所有这些都遵从 Spring 的通用事务和 DAO 异常层次结构。
* Web模块（Spring Web）：提供基本的面向Web的综合性，提供了Web应用的一些抽象封装。所以，Spring框架支持与Jakarta Struts、Webwork等的集成。Web模块还简化了处理多部分请求以及将请求参数绑定到域对象的工作。
* Spring MVC框架：通过策略接口，MVC框架变成为高度可配置的，MVC容纳了大量视图技术，其中包括 JSP、Velocity、Tiles、iText和POI 。[9]

## 2.3 Hibernate

### 2.3.1 Hibernate简介

Hibernate是一种比较彻底的Java对象关系映射（ORM）工具，是目前JAVA数据库操作方式中最好的解决方案之一，支持使用各种Java思想来实现对象。其在JDBC基础上进行了轻量级的封装，使得开发人员只需要少量代码便可以完成数据持久化操作，大幅度减少开发时人工使用SQL和JDBC处理数据的时间。同时Hibernate还拥有一种功能强大的查询语言HQL，便于开发人员掌握，是完全面向对象的，查询的是持久化对象。最重要的一点，就是Hibernate支持大部分主流关系型数据库，支持父子关系、事务处理、继承等。

随着产品线的逐渐完善，Hibernate增加了更多的工具套件，并支持.net平台。Hibernate的轻量级 ORM 模型逐步确立了在Java ORM架构中领导地位，甚至取代复杂的EJB模型而成为事实上的Java ORM工业标准，并且其中许多设计均被J2EE 标准组织吸纳而成为最新EJB3.0规范的标准[10]。

### 2.3.2 Hibernate优势特点

Hibernate与CMP的实现方式完全不同。这种持久化服务是一个由业务逻辑调用的接口，而它的一大优势正在于开发者可以把被持久化的对象实现成脱离这个持久化环境依然可以运行的类。

多数开发人员会用JDBC来和数据库进行通信，并通过DAO模式进行改善和提高，但这种方式在大型应用程序中不易操作使用，而且维护起来相当困难。Hibernate这种持久化框架则无需任何容器，能够提供简单易用并符合ODMG3-style的API,也解决了JDO的诸多缺陷。作为一个良好的ORM，它有下面几个特点：

（1）开源免费的License，可在需要的时候研究、改写源代码，进行功能的定制。

（2）透明提供对象与关系数据库的映射，以统一的接口方式支持多种数据库。

（3）轻量级，避免引入过多复杂问题，容易调试。

（4）缓存机制，针对数据库操作可大大减少。

（5）具有可扩展性，且开发者活跃，产品具有稳定的发展保障。

总之，Hibernate是一个优秀的开放源代码的Java对象持久层轻量级封装框架，既可以用来在Java应用程序中取代大部分JDBC代码，也可以整合到J2EE系统中作为持久层框架。

2.3.3 Hibernate结构

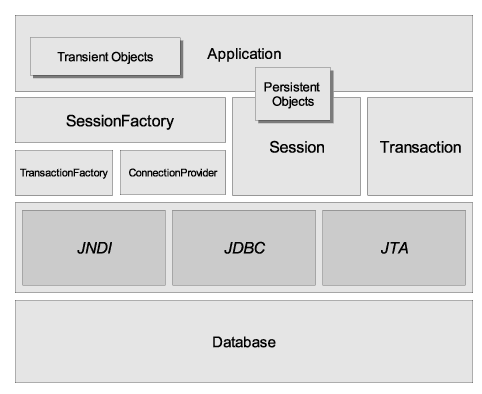


图2.3 Hibernate结构图

图2.4显示Hibernate的一种复杂体系结构图，它作为应用层与数据库层的中间层，通过加载数据库连接、实体类、映射信息等配置信息，创建持久化的对象来同步数据，大大提升了数据库操作效率。

## 2.4 本章小结

本章介绍了智造链系统中使用到的相关技术的概述，主要为jBPM工作流框架，Spring的框架及结构，以及数据持久化工具Hibernate的结构和优点。

# 第三章 智造链系统需求分析与概要设计

## 3.1 智造链系统整体概述

本项目是为了实现对智造链公司的整个客户下单的复杂业务流程进行管理，将所有的业务流程都集成在一个较独立的线上系统中，方便市场主管、市场专员等公司职工进行结构化、流程化的管理和信息交互，摒除原来的纯手工的、低效率的订单管理方式。系统为所有的参与到系统中的公司员工提供账号，根据不同员工的职能和需要处理的任务，为不同的角色提供各自的用户页面，将所有的任务分配到具体的角色，确保对整个流程能够进行高效的管理。

### 3.1.1 系统用户角色

本系统的使用者为公司内部的员工，主要角色有：市场主管、市场专员、设计主管、采购主管、生产主管、财务主管、物流主管、质检主管和系统管理员。市场主管和市场专员主要负责联系客户和下订单，以及一系列和客户相关的任务，其中市场主管可以审核报价，负责对订单的审查等工作；设计主管主要负责和服装设计相关的任务，包括设计验证、设计样衣版型、设计生产验证和设计生产版型等；采购主管主要负责和原料采购有关的任务，包括采购验证、采购成本核算、样衣采购、采购确认和生产采购等；生产主管主要负责和服装的生产相关的任务，包括生验证、生产成本核算、样衣生产和批量生产等；财务主管主要负责和公司财务有关的任务，包括样衣费、首定金和尾款金的确认等；物流主管主要负责和物流相关的任务，包括样衣的收取和发货、产品的入库和发货等；质检主管主要负责对批量生产的服装进行质量检查；系统管理员主要负责系统的维护工作，包括系统日志的查看和故障的修复等，以及对系统用户的维护和管理工作等。目前本项目的1.0版本只提供公司内部员工使用，不开放给外部客户使用。

### 3.1.2 系统功能性需求



图3.1 智造链系统模块图

智造链系统模块划分如图3.1所示。

根据用户角色的不同，将系统的功能需求分为九个模块。

市场部模块：包括添加客户、查看客户、客户下单、修改询单、合并报价、审核报价、报价商定、修改报价、商定合同、修改合同、签订合同、订单回访等主要功能。

设计部模块：包括设计验证、录入样衣版型、设计生产验证、生产版型确认等主要功能。

采购部模块：包括采购验证、采购成本核算、样衣采购、采购确认和生产采购等主要功能。

生产部模块：包括生产验证、生产成本核算、样衣生产和批量生产等主要功能。

财务部模块：包括样衣费确认、首定金确认和尾款金确认等主要功能。

物流部模块：包括样衣收取、样衣发货、产品入库和产品发货等主要功能。

质检部模块：包括质量检查功能。

系统管理员模块：包括添加用户、查看用户信息、修改用户信息和删除用户等主要功能。

人事部模块：包括添加员工、查看员工、修改和删除员工信息等功能。

### 3.1.3 系统非功能性需求

**可用性**

系统将不同角色的功能分开，使用户能清楚的明白自己需要做什么，专注于自己的任务，不必分心去寻找界面的菜单或者理解界面的结构与图标的含义。

系统在用户进行操作后给出相应的提示信息，同时当用户操作有误是也给出相信的错误信息，使用户容易理解，降低使用系统的难度。

软件界面要符合一般用户的操作习惯，操作动作简单，用户学习操作的时间短。

**支持性**

系统需要能在Windows、Mac OS等操作系统上运行，在不同浏览器如IE、Firefox、Chrome等多种浏览器上运行效果良好，同时部分功能在移动端浏览器上能够良好运行。

**安全性**

系统需要保证用户、公司客户和订单信息等重要资料的安全性。

**系统性能**

系统需要在用户操作后的1秒内及时给出反馈，可上传大小为10M的文件。

### 3.1.4 系统运行环境

1. 软件环境

Web服务器：Tomcat 7.x

数据库：Mysql 5.x

JDK：7.0以上

浏览器：IE，Mozilla Firefox，Chrome等

操作系统：Windows，Mac OS

2. 硬件环境

服务器：内存2G以上，硬盘100G以上，CPU速度2GHz以上

客户端：内存1G以上，有浏览器

## 3.2 智造链系统设计部模块的需求分析

### 3.2.1 设计部模块用例描述

设计部模块包括四个主要用例，分别是设计验证、录入样衣版型、设计生产验证、生产版型确认。

1. 设计验证：

在客户下单之后，需要设计部、采购部和生产部对订单信息进行验证，如果全部通过，则可以进入下一步，不通过则要继续修改询单。设计验证由设计主管进行操作，系统提供详细的订单信息，设计主管可以进行同意或拒绝操作并填写意见。

1. 录入样衣版型：

当收到客户的样衣制作金之后，设计部需要进行样衣的版型设计。系统提供样衣版型CAD文件的上传功能，设计主管需在样衣版型设计好之后并将样衣版型CAD文件上传到服务器并提交。

1. 设计生产验证：

在市场部确认完加工单之后，进行采购部和设计部确认，若都通过，则进入下一步签订合同。设计部确认由设计部主管进行操作，系统提供已经录入的样衣版型CAD文件的下载，如果有需要修改的地方则在修改完成之后重新上传CAD文件并提交，如果没有需要修改的地方则不需要再上传文件。

1. 生产版型确认：

在用户交了30%的首定金之后，进行最后一次设计部的生产版型确认，由设计部主管进行操作，系统提供已经录入的CAD文件的下载，如果有需要修改的地方则在修改完成之后重新上传CAD文件并提交，如果没有需要修改的地方则不需要再上传文件。如果设计部和采购部都通过则进入下一步批量生产。

### 3.2.2 设计部模块用例图

智造链系统设计部模块的用例图如图3.2所示，主要包括设计验证、录入样衣版型、设计生产验证以及生产版型确认等四个用例。

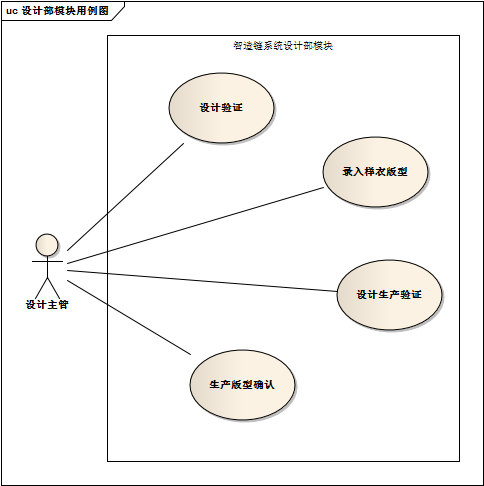


图3.2 设计部模块用例图

### 3.2.3 设计部模块实体关系分析

图3.3为智造链系统设计部模块的实体关系图，设计部模块涉及的实体包括员工实体、订单实体、客户实体、CAD文件实体等。

每个订单有且仅有一位客户，故订单与客户的对应关系为1:1。

每个订单有且仅有一个CAD文件，故订单与CAD文件的对应关系为1:1。

员工可以作为订单的创建或者修改人员，故员工与订单之间存在创建或者修改的联系。

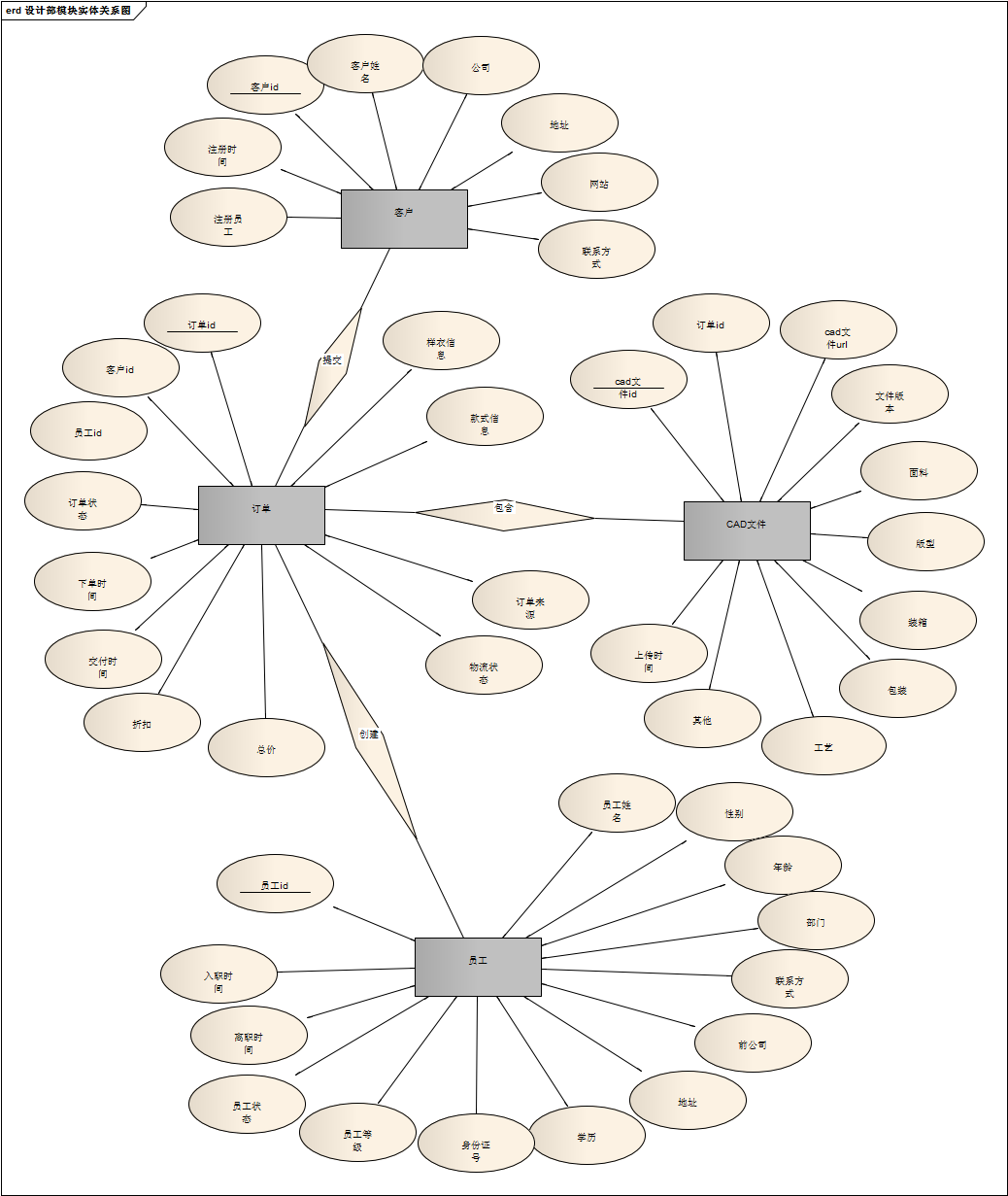


图3.3 设计部模块实体关系图

## 3.3 智造链系统生产部模块的需求分析

### 3.3.1 生产部模块用例描述

生产部模块包括四个主要用例，分别是生产验证、生产成本核算、样衣生产和批量生产。

1. 生产验证：

在客户下单之后，需要设计部、采购部和生产部对订单信息进行验证，如果全部通过，则可以进入下一步，不通过则要继续修改询单。生产验证由生产主管进行操作，系统提供详细的订单信息，生产主管可以进行同意或拒绝操作并填写意见。

1. 生产成本核算：

通过验证之后，生产部需要对生产成本进行核算。系统提供详细的订单信息，生产主管将各项生产成本费用计算好之后填入相应的表单内，并进行提交。

1. 样衣生产：

在采购确认和录入样衣版型数据之后，需要进行样衣生产。系统提供详细的订单数据，在样衣生产完成后，由生产主管根据样衣生产是否成功，选择加工完成或者加工失败。

1. 批量生产：

当设计部生产版型确认和采购部生产原料采购都完成之后，进行批量生产。系统提供详细的计划生产数量，由生产主管在批量生产完成后填写详细的实际生产数量，并根据批量生产是否成功选择加工完成或者加工失败。

### 3.3.2 生产部模块用例图

智造链系统生产部模块的用例图如图3.4所示，主要包括生产验证、生产成本核算、样衣生产和批量生产等四个用例。

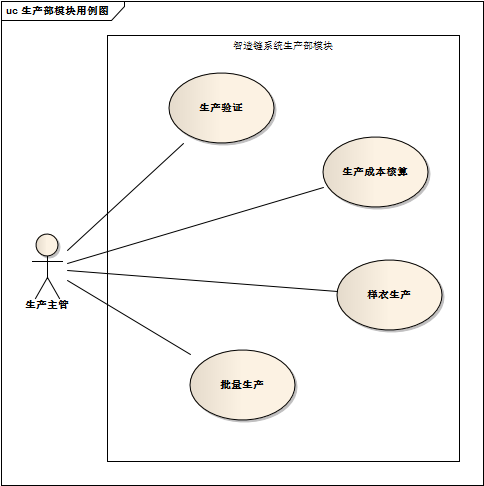


图3.4 生产部模块用例图

### 3.3.3 生产部模块实体关系分析

图3.5为智造链系统生产部模块的实体关系图，生产部模块涉及的实体包括员工实体、订单实体、客户实体、加工单实体等。

每个订单有且仅有一位客户，故订单与客户的对应关系为1:1。

每个订单中有多个加工单，故订单与加工单的对应关系为1:N。

员工可以作为订单的创建或者修改人员，故员工与订单之间存在创建或者修改的联系。

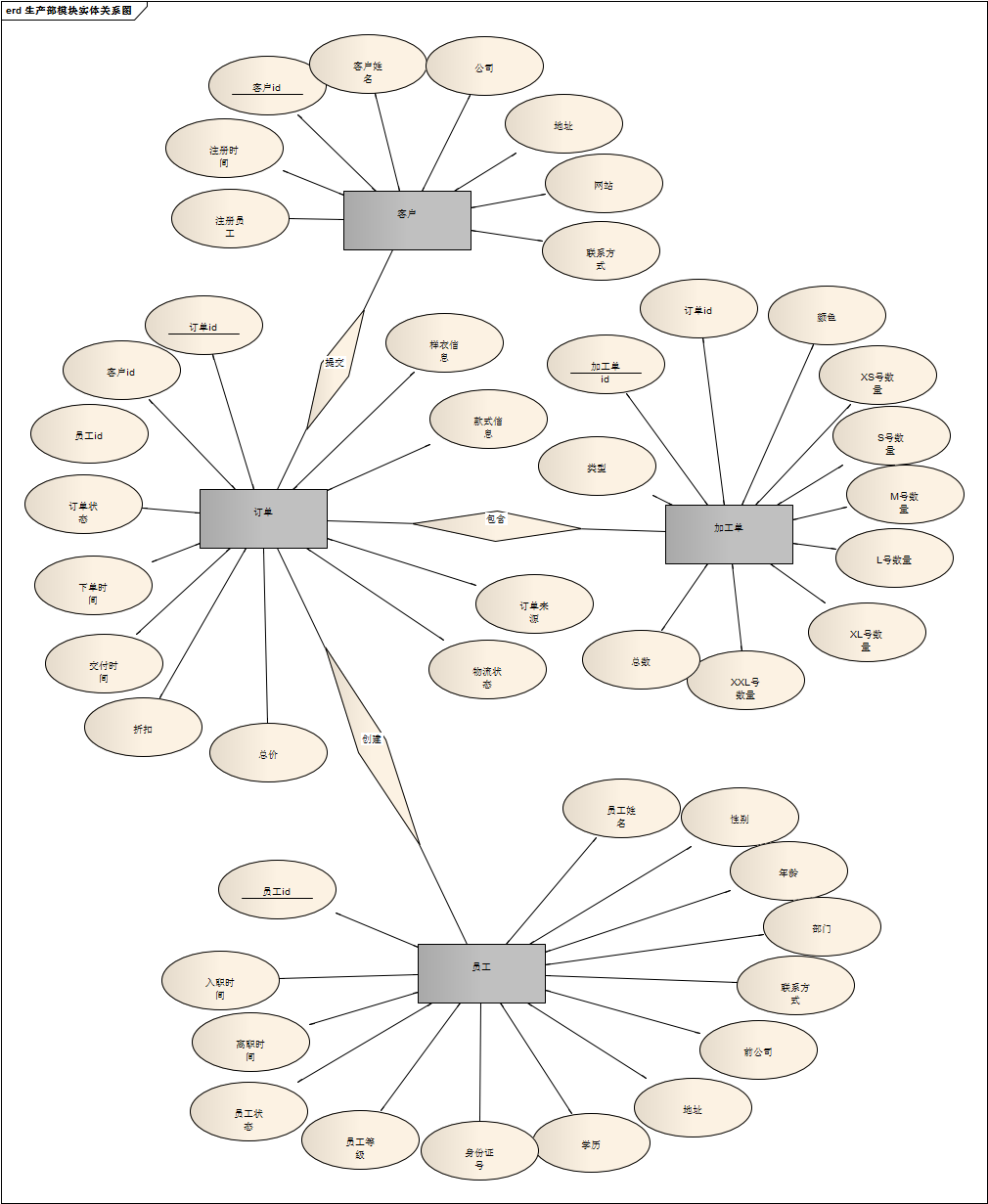


图3.5 生产部模块实体关系图

## 3.4 智造链系统质检部模块的需求分析

### 3.4.1 质检部模块用例描述

质检部模块包括一个主要用例，即质量检查。

质量检查：

当批量生产完成之后，质检部需要进行质量检查。系统提供详细的生产数量，由质检主管在质量检查完成后填写详细的通过检查数量与不通过检查数量。质量检查完成后进入下一步入库。

### 3.4.2 质检部模块用例图

智造链系统设计部模块的用例图如图3.6所示，主要为质量检查用例。

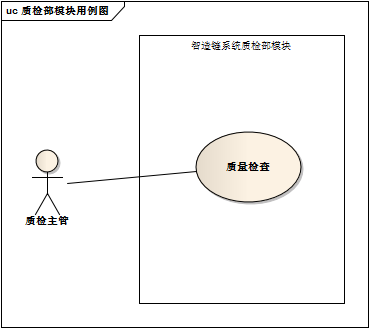


图3.6 质检部模块用例图

### 3.4.3 质检部模块实体关系分析

因质检部工作即对生产部批量生产的产品进行质检，所以质检部的实体关系同生产部一样，即都为图3.5生产部模块实体关系图。质检部模块涉及的实体包括员工实体、订单实体、客户实体、加工单实体等。

每个订单有且仅有一位客户，故订单与客户的对应关系为1:1。

每个订单中有多个加工单，故订单与加工单的对应关系为1:N。

员工可以作为订单的创建或者修改人员，故员工与订单之间存在创建或者修改的联系。

## 3.5 智造链系统设计部模块的概要设计

本章主要介绍了智造链系统设计部模块的程序结构设计、功能组件划分以及相关接口设计。

### 3.5.1 设计部模块程序结构设计

智造链系统的设计部模块分为五层，分别是View层、Controller层、Service层、DAO层和数据库层。从Controller层到数据库层只能一层一层往下调用，不能反向调用，将前端、逻辑处理和数据库层分离，使系统的逻辑更清晰，降低了系统的耦合度并提高了重用性。

系统的框架结构如图3.7所示。用户通过View层与系统进行交互，当用户执行某个操作时，用户的操作信息将被传递到Controller层，Controller层对接收的信息进行初步的处理，然后调用相应的Service层的方法进行处理。在Service层，系统对信息进行逻辑处理，对于需要持久化的信息，Service层会调用对应的DAO层的方法，将数据同步到数据库。DAO处理完成后将结果返回给Service层，同样Service层也将信息返回给上一层，一直到View层并显示给用户。



图3.7 系统程序结构图

### 3.5.2 设计部模块功能组件划分

如图3.8所示，智造链系统由多个模块组件构成，包括配置模块、数据库模块、文件操作模块、邮件提醒模块、用户管理模块、订单管理模块、流程管理模块以及访问控制模块。

设计部模块涉及的功能组件有文件操作组件、流程管理组件和数据库组件。



图3.8 系统功能组件划分图

### 3.5.3 设计部模块接口设计

智造链系统设计部模块的接口主要包括Service层和DAO层两层，Service层接口为DesignService接口，DAO层接口为IDesignCadDAO接口。

DesignService接口由DesignServiceImpl类实现，其具体接口定义如表3.1所示。该接口主要提供了查看任务列表、显示任务详情和提交任务的方法接口。

表3.1 DesignService接口方法表

|  |  |
| --- | --- |
| 方法名 | 描述 |
| getVerifyDesignList | 获得设计验证的任务列表 |
| getVerifyDesignDetail | 获取一个设计验证任务的详细信息 |
| verifyDesignSubmit | 提交一个设计验证任务 |
| getUploadDesignList | 获得录入样衣版型的任务列表 |
| getUploadDesignDetail | 获取一个录入样衣版型任务的详细信息 |
| uploadDesignSubmit | 提交一个录入样衣版型任务 |
| getModifyDesignList | 获得设计生产验证的任务列表 |
| getModifyDesignDetail | 获取一个设计生产验证任务的详细信息 |
| modifyDesignSubmit | 提交一个设计生产验证任务 |
| getConfirmDesignList | 获得生产版型确认的任务列表 |
| getConfirmDesignDetail | 获取一个生产版型确认任务的详细信息 |

IDesignCadDAO接口由DesignCadDAO类实现，其接口定义如表3.2所示。该接口主要提供了和cad文件信息操作有关的方法接口，包括添加、更新、删除和查找cad文件信息等。其中查找包括了通过一些重要字段进行查找的方法接口。

表3.2 DesignCadDAO接口方法表

|  |  |
| --- | --- |
| 方法名 | 描述 |
| save | 添加cad文件信息 |
| delete | 删除cad文件信息 |
| findById | 通过id查找cad文件信息 |
| findByExample | 通过实例查找cad文件信息 |
| findByProperty | 通过特定属性查找cad文件信息 |
| findByOrderId | 通过订单id查找cad文件信息 |
| findByCadUrl | 通过cad的url查找cad文件信息 |
| findByCadVersion | 通过cad版本号查找cad文件信息 |
| findAll | 查找所有cad文件信息 |
| merge | 更新cad文件信息 |

## 3.6 智造链系统生产部模块的概要设计

本章主要介绍了智造链系统生产部模块的程序结构设计、功能组件划分以及相关接口设计。

### 3.6.1 生产部模块程序结构设计

智造链系统的生产部模块的程序结构和设计部相似，都是分为View层、Controller层、Service层、DAO层和数据库层。系统结构参考图3.7系统程序结构图。

### 3.6.2 生产部模块功能组件划分

生产部模块涉及的功能组件有订单管理组件、流程管理组件和数据库组件。

### 3.6.3 生产部模块接口设计

智造链系统生产部模块的接口主要包括Service层和DAO层两层，Service层接口为ProduceService接口，DAO层接口为IProduceDAO接口。

ProduceService接口由ProduceServiceImpl类实现，其具体接口定义如表3.3所示。该接口主要提供了查看任务列表、显示任务详情、提交任务和获取加工单的方法接口。

表3.3 ProduceService接口方法表

|  |  |
| --- | --- |
| 方法名 | 描述 |
| getVerifyProduceList | 获得生产验证的任务列表 |
| getVerifyProduceDetail | 获取一个生产验证任务的详细信息 |
| verifyProduceSubmit | 提交一个生产验证任务 |
| getComputeProduceCostList | 获得生产成本核算的任务列表 |
| getComputeProduceCostInfo | 获取一个生产成本核算任务的详细信息 |
| computeProduceCostSubmit | 提交一个生产成本核算型任务 |
| getProduceSampleList | 获得样衣生产的任务列表 |
| getProduceSampleDetail | 获取一个样衣生产任务的详细信息 |
| produceSampleSubmit | 提交一个样衣生产任务 |
| getProduceList | 获得批量生产的任务列表 |
| getProduceDetail | 获取一个批量生产任务的详细信息 |
| pruduceSubmit | 提交一个批量生产任务 |
| getProduceList | 获得加工单列表 |

IProduceDAO接口由ProduceDAO类实现，其接口定义如表3.4所示。该接口主要提供了和加工单操作有关的方法接口，包括添加、更新、删除和查找加工单信息等。其中查找包括了通过一些重要字段进行查找的方法接口。

表3.4 IProduceDAO接口方法表

|  |  |
| --- | --- |
| 方法名 | 描述 |
| save | 添加加工单信息 |
| delete | 删除加工单信息 |
| findById | 通过id查找加工单信息 |
| findByExample | 通过实例查找加工单信息 |
| findByProperty | 通过特定属性查找加工单信息 |
| findProduceByOrderId | 通过订单id查找加工单信息 |
| findSampleProduceByOrderId | 通过订单id查找样衣加工单信息 |
| findAll | 查找所有加工单信息 |
| merge | 更新加工单信息 |
| deleteProduceByProperty | 通过特定属性删除加工单信息 |
| deleteSampleProduceByProperty | 通过特定属性删除样衣加工单信息 |

## 3.7 智造链系统质检部模块的概要设计

本章主要介绍了智造链系统质检部模块的程序结构设计、功能组件划分以及相关接口设计。

### 3.7.1 质检部模块程序结构设计

智造链系统的质检部模块的程序结构和设计部相似，都是分为View层、Controller层、Service层、DAO层和数据库层。系统结构参考图3.7系统程序结构图。

### 3.7.2 质检部模块功能组件划分

质检模块涉及的功能组件有订单管理组件、流程管理组件和数据库组件。

### 3.7.3 质检部模块接口设计

智造链系统质检部模块的接口主要包括Service层和DAO层两层，Service层接口为QualityService接口，DAO层接口为IProduceDAO接口。

QualityService接口由QualityServiceImpl类实现，其具体接口定义如表3.5所示。该接口主要提供了查看任务列表、显示任务详情和提交任务的方法接口。

表3.5 QualityService接口方法表

|  |  |
| --- | --- |
| 方法名 | 描述 |
| getCheckQualityList | 获得质量检查的任务列表 |
| getCheckQualityDetail | 获取一个质量检查任务的详细信息 |
| checkQualitySubmit | 提交一个质量检查任务 |

IProduceDAO接口由ProduceDAO类实现，已在3.6.3节中描述过，此处不再赘述。

## 3.8本章小结

本章首先介绍了智造链系统的整体概述，描述了其模块划分，接着使用用例图、系统顺序图和活动图等UML图对该系统进行需求分析，包括功能性需求和非功能性需求，使用ERD图描述了系统的数据库设计。另外，本章还介绍了智造链系统各模块之间的接口。

# 第四章 智造链系统设计部模块、生产部模块与质检部模块的详细设计与实现

## 4.1 设计部模块、生产部模块和质检部模块概述

### 4.1.1 设计部模块概述

设计部的任务在本项目中全部由设计主管来进行操作，设计主管主要负责与产品设计相关的任务，包括设计验证、录入样衣版型、设计生产验证和生产版型确认。制作样衣前，设计主管需要完成设计验证和录入样衣版型任务。设计验证是对订单进行验证，而录入样衣版型则是将设计好的样衣版型的cad文件上传到服务器。设计生产验证和生产版型确认都是相对于产品的批量生产。在设计生产验证中，设计主管需要核对样衣版型版型，若要修改则要上传新的版型cad文件。而生产版型确认则是对版型的最后一次确认，没有问题即进行批量生产。这四个任务即设计主管在本系统中的主要任务。

经过详细的分析，将设计部模块按流程分为四部分页面，即设计验证、录入样衣版型、设计生产验证和生产版型确认，每部分又包括任务列表和任务详情两个页面。

### 4.1.2 生产部模块概述

生产部的任务在本项目中全部由生产主管来进行操作，生产主管主要负责与产品生产相关的任务，包括生产验证、生产成本核算、样衣生产和批量生产。在制作样衣前，生产主管需要完成生产验证、生产成本核算和样衣生产这三个任务。生产验证是对订单进行验证；生产成本核算则是对生产过程中的各种成本进行核算，包括裁剪、缝制、设计等费用和其他一些费用；样衣生产则是在样衣生产结束后，生产主管根据样衣生产成功与否，选择样衣生产是否成功。批量生产则是在批量生产结束之后，生产主管将实际的生产数量与计划生产数量进行核对，将数据填入表单并提交。这四个任务即生产主管在本系统中的主要任务。

经过详细的分析，将生产部模块按流程分为四部分页面，即生产验证、生产成本核算、样衣生产和批量生产，每部分又包括任务列表和任务详情两个页面。

### 4.1.3 质检部模块概述

质检部在本系统中只有质量检查一个任务，由质检主管负责进行操作。质量检查是在质检部完成质量检查后，由质检主管将质量检查结果填入相应的表单中并提交。经过分析，将质检部模块分为两个页面，即任务列表和任务详情。

## 4.2 设计部模块的详细设计

### 4.2.1 设计部模块详细类图

如图4.1所示，设计部模块涉及到了controller、service、serviceImpl、dao和daoImpl五个包中的部分类。DesignController类负责与前端页面的交互，当数据传到DesignController后，调用DesignService来完成请求的操作，DesignService则再调用相应的dao类来与数据库交互。DesignService接口由DesignServiceImpl类来实现，dao接口则由对应的daoImpl中的类来实现。

### 4.2.2 设计部模块顺序图

图4.2所示为设计部模块顺序图，以上传样衣版型为例，主要由设计主管、系统前端、服务器端以及数据库端组成。

设计主管首先选择某个流程，系统前端页面会通过服务器发送请求查找数据库并返回相应的信息给前端页面，前端页面显示任务列表，设计主管再选择一条任务进入任务详情页面，服务器通过数据库查找到详细的任务信息并显示给设计主管，设计主管进行操作后，新的数据通过前端传送到服务器并同步到数据库，任务完成，流程进入下一步。

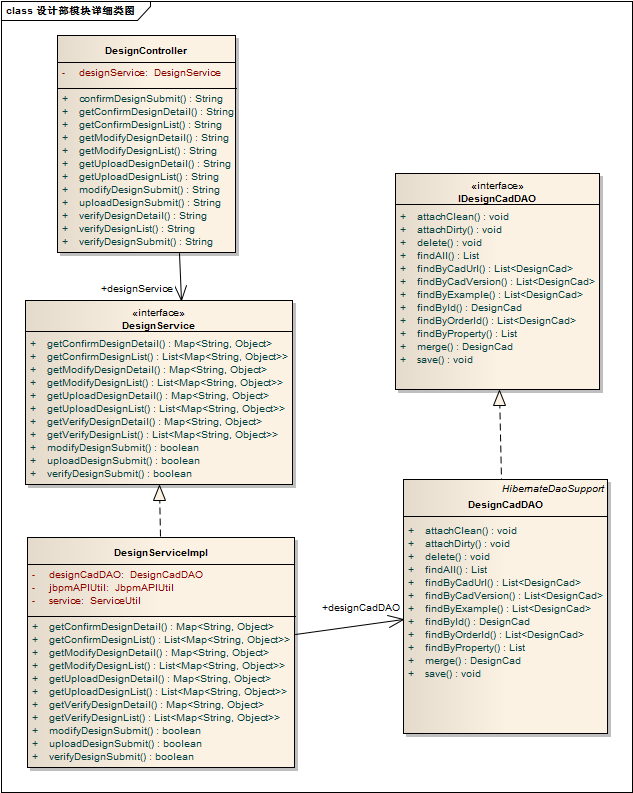


图4.1 设计部模块详细类图

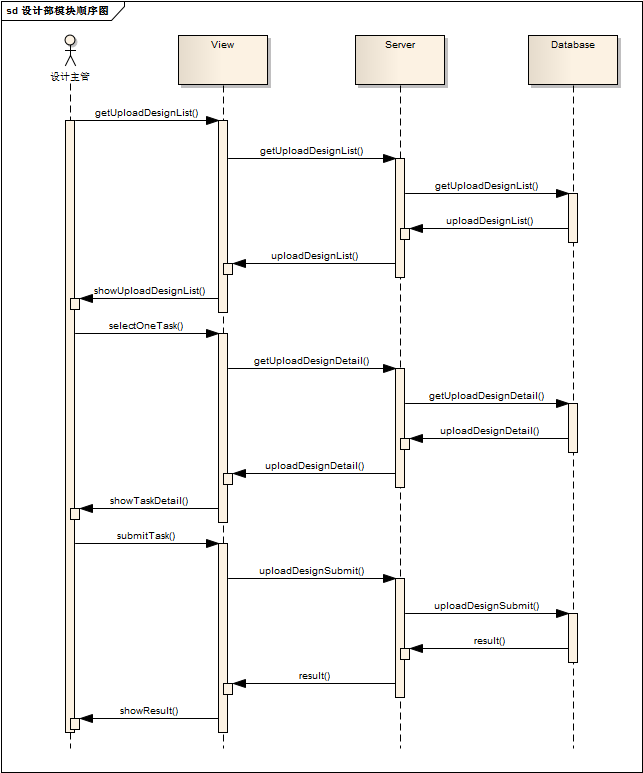


图4.2 设计部模块顺序图

### 4.2.3 设计部模块关键数据表设计

生产部模块涉及一个关键数据表，即加工单信息表。

在设计主管上传cad文件后，数据库中即存在一条cad文件信息的记录。由于每个订单只存在一条cad信息，因此订单信息与cad文件信息的对应关系为1:1，cad信息表中包含一个订单id的外键。Cad文件信息表的详细描述见表4.1。

表4.1 cad文件信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| cad\_id | int(11) | 用于区分每个cad文件信息的唯一编号 |
| cad\_box | varchar(225) |  |
| cad\_fabric | varchar(225) |  |
| cad\_other | varchar(225) |  |
| cad\_package | varchar(225) |  |
| cad\_tech | varchar(225) |  |
| cad\_url | varchar(225) | cad文件的url路径 |
| cad\_version | smallint(6) | cad文件的版本号 |
| cad\_version\_data | varchar(225) | cad文件的版本信息 |
| order\_id | int(11) | 订单编号 |
| upload\_time | datetime | 文件上传时间 |

## 4.3 生产部模块的详细设计

### 4.3.1 生产部模块详细类图

如图4.3所示，生产部模块涉及到了controller、service、serviceImpl、dao和daoImpl五个包中的部分类。ProduceController类负责与前端页面的交互，当数据传到ProduceController后，调用ProduceService来完成请求的操作，ProduceService则再调用相应的dao类来与数据库交互。ProduceService接口由ProduceServiceImpl类来实现，dao接口则由对应的daoImpl中的类来实现。

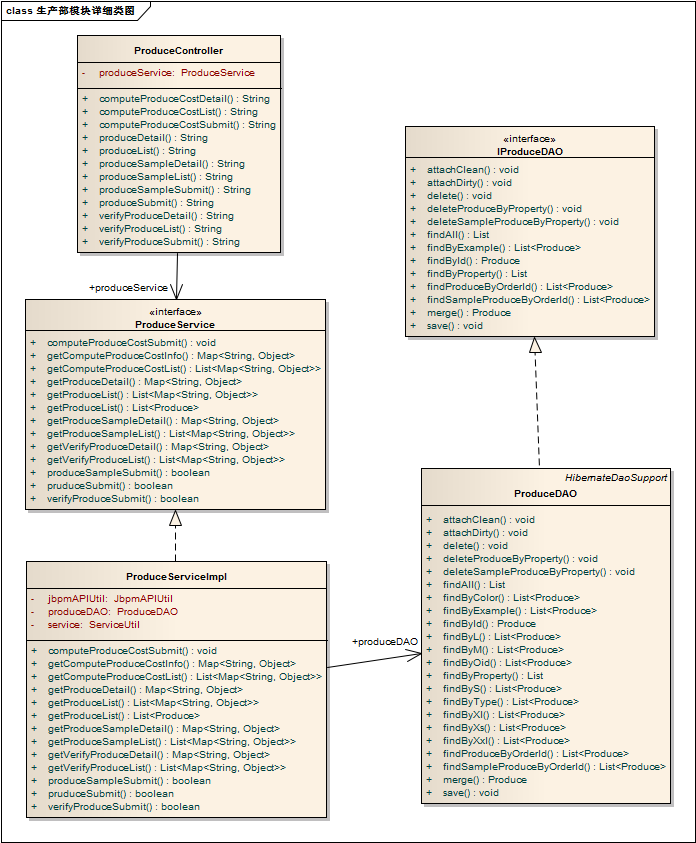


图4.3 生产部模块详细类图

### 4.3.2 生产部模块顺序图

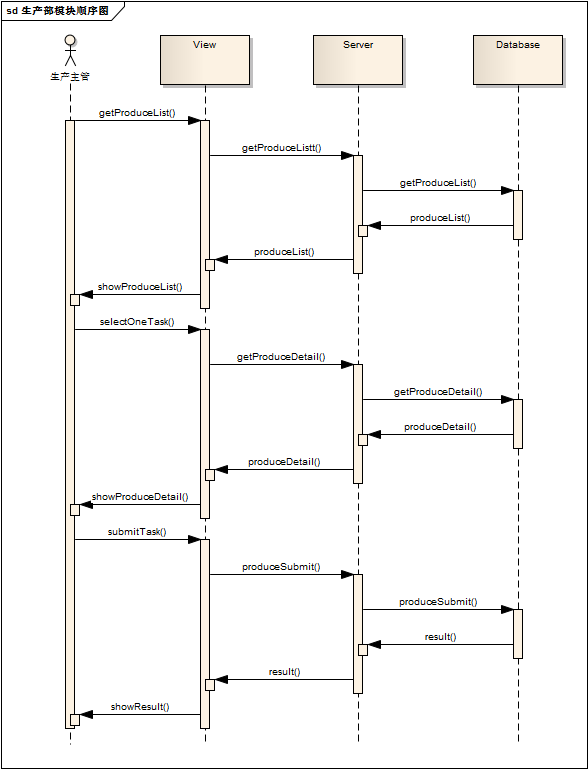


图4.4 生产部模块顺序图

图4.4所示为生产部模块顺序图，以批量生产为例，主要由生产主管、系统前端、服务器端以及数据库端组成。

首先由生产主管选择某个流程，系统前端页面会通过服务器发送请求查找数据库并返回相应的信息给前端页面，前端页面显示任务列表，生产主管再选择一条任务进入任务详情页面，服务器通过数据库查找到详细的任务信息并显示给生产主管，生产主管进行操作后，新的数据通过前端传送到服务器并同步到数据库，任务完成，流程进入下一步。

### 4.3.3 生产部模块关键数据表设计

生产部模块涉及一个关键数据表，即加工单信息表。

生产主管在完成批量生产任务时需要填写实际生产数量表，即加工单信息表。由于每个订单会存在多条加工单信息记录，因此订单信息表与加工单信息表的对应关系为N:1。加工单信息表中包含一个订单id的外键。加工单信息表的详细信息见表4.2。

表4.2 加工单信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| pid | int(11) | 用于区分每个加工单信息的唯一编号 |
| color | varchar(225) | 此加工单的颜色信息 |
| L | int(11) | 加工单中L码的产品数量 |
| M | int(11) | 加工单中M码的产品数量 |
| S | int(11) | 加工单中S码的产品数量 |
| XL | int(11) | 加工单中XL码的产品数量 |
| XS | int(11) | 加工单中XS码的产品数量 |
| XXL | int(11) | 加工单中XXL码的产品数量 |
| produceAmount | int(11) | 加工单中的产品总数量 |
| type | varchar(225) | 加工单类型 |
| oid | int(11) | 订单编号 |

## 4.4 质检部模块的详细设计

### 4.4.1 质检部模块详细类图

如图4.5所示，质检部模块涉及到了controller、service、serviceImpl、dao和daoImpl五个包中的部分类。QualityController类负责与前端页面的交互，当数据传到QualityController后，调用QualityService来完成请求的操作，QualityService则再调用相应的dao类来与数据库交互。QualityService接口由QualityServiceImpl类来实现，dao接口则由对应的daoImpl中的类来实现。

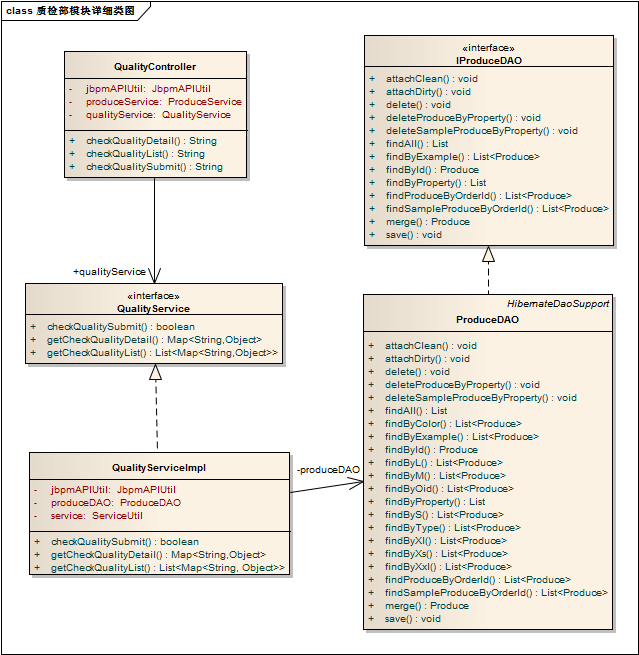


图4.5 质检部模块详细类图

### 4.4.2 质检部模块顺序图

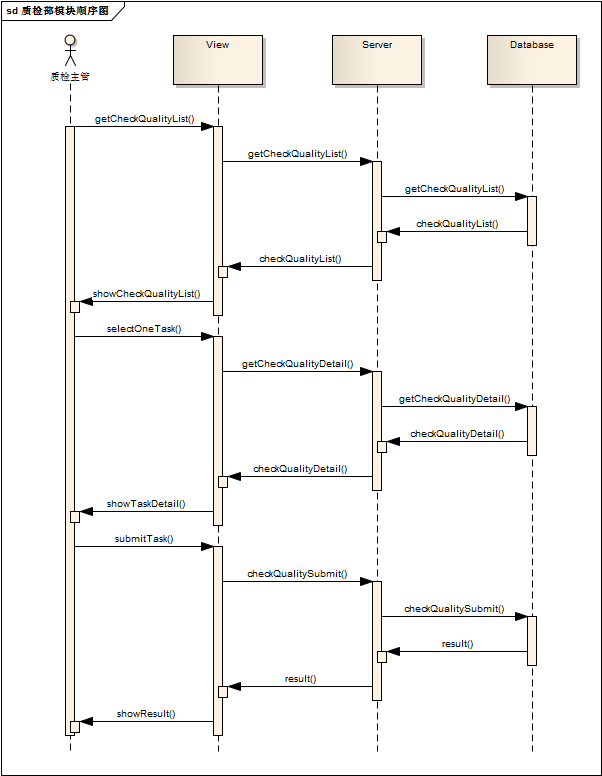


图4.6 质检部模块顺序图

图4.6所示为质检部模块顺序图，主要由质检主管、系统前端、服务器端以及数据库端组成。

首先由质检主管选择某个流程，系统前端页面会通过服务器发送请求查找数据库并返回相应的信息给前端页面，前端页面显示任务列表，质检主管再选择一条任务进入任务详情页面，服务器通过数据库查找到详细的任务信息并显示给质检主管，质检主管进行操作后，新的数据通过前端传送到服务器并同步到数据库，任务完成，流程进入下一步。

### 4.4.3 质检部模块关键数据表设计

质检部模块涉及一个关键数据表，即加工单信息表。

质检主管在进行质量检查时，需要将检查后的产品信息填入相应的质检表中，即加工单信息表。这里的加工单信息表和4.3.3节中描述的加工单信息表相同，此处不再赘述。

## 4.5 设计部模块的实现

设计部模块主要包括设计验证、录入样衣版型、设计生产验证和生产版型确认等操作。本节描述了设计部模块各操作的具体实现细节，以及相关页面截图。

### 4.5.1 设计验证

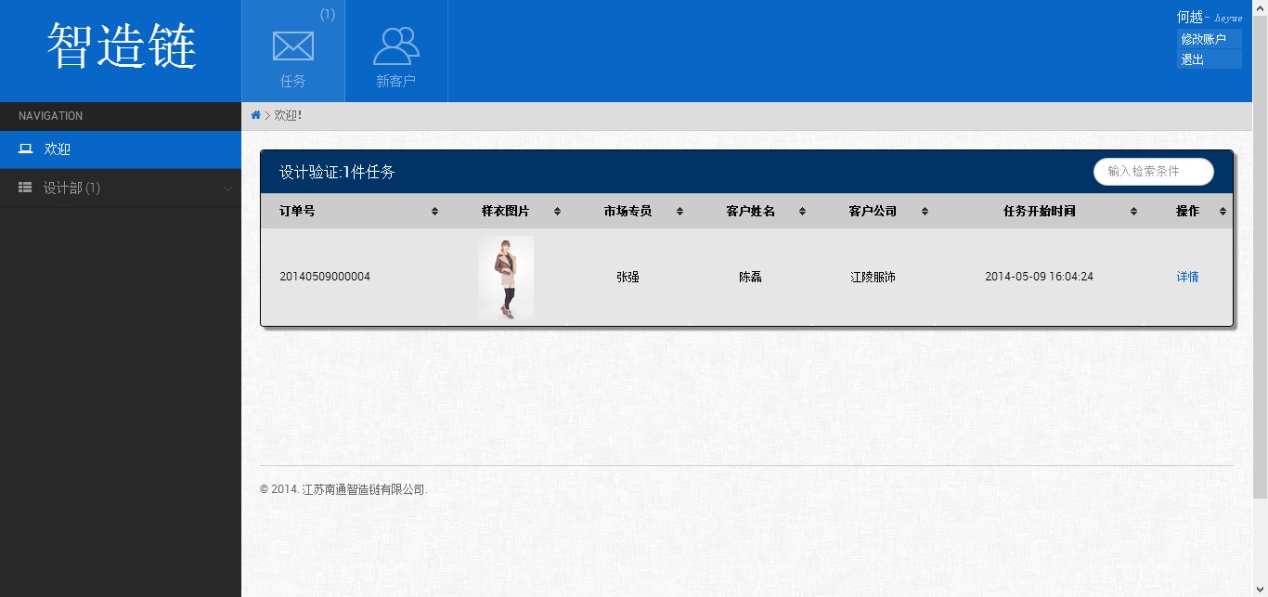


图4.7 设计部模块设计验证任务列表页面

如图4.7所示，进行设计验证首先会显示一个设计验证的任务列表，列表中显示每条任务的一些简单信息，包括订单号、样衣图片、市场专员、客户等关键信息。每条任务右侧有详情按钮，可点击进入查看任务详情。

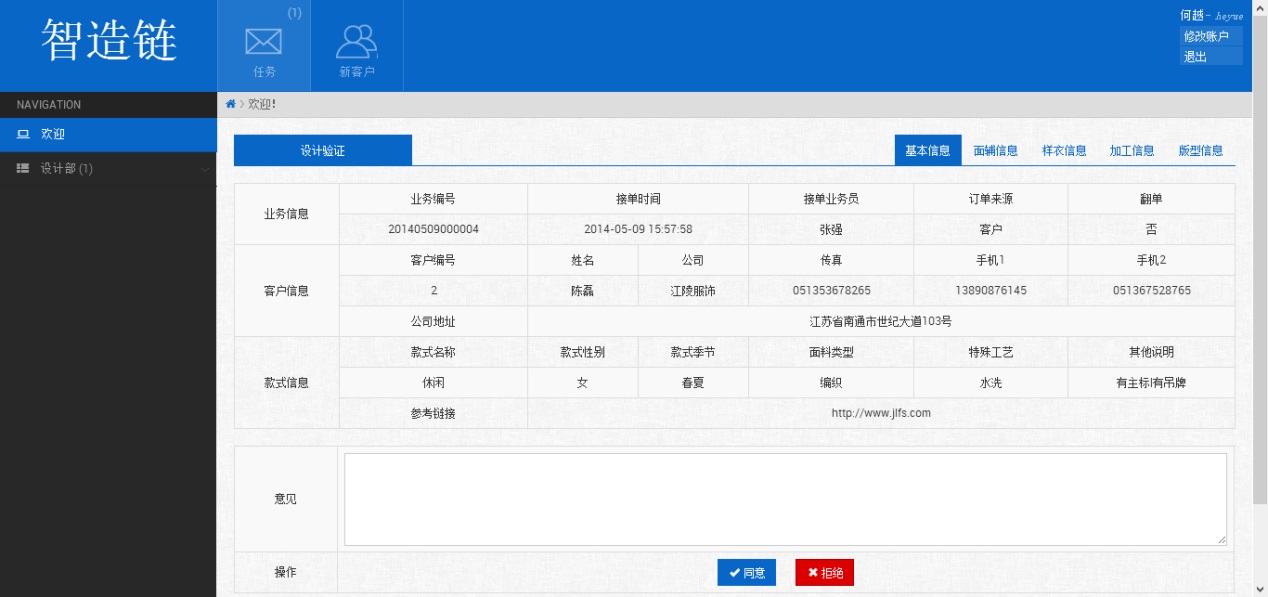


图4.8 设计部模块设计验证任务详情页面

任务详情页面如图4.8所示。在右上方可看出，任务详情页面包含五个小的标签页，分别为基本信息、面辅信息、样衣信息、加工信息和版型信息，在页面下方有意见栏和两个操作按钮，在设计主管查看完订单的详细信息后，可以填写意见并选择同意或者拒绝操作，后台会进行相应的处理并将数据同步到数据库。

后台数据处理如图4.9所示，data用来存放流程中需要的参数，包括设计验证是否通过和填写的意见，然后调用封装好的jbpmAPIUtil中的completeTask来完成任务，需要传入的参数有任务id、流程参数和进行操作的角色。

|  |
| --- |
| public boolean verifyDesignSubmit(long taskId, boolean result, String comment) {  //data中存放流程中需要的参数  Map<String, Object> data = new HashMap<>();  data.put(RESULT\_DESIGN, result);//设计验证的结果是同意还是拒绝  data.put(RESULT\_DESIGN\_COMMENT, comment);//存放意见  try {  //完成任务，传入任务id、流程参数以及操作角色等参数  jbpmAPIUtil.completeTask(taskId, data, ACTOR\_DESIGN\_MANAGER);  return true;  } catch (InterruptedException e) {  e.printStackTrace();  return false;  }  } |

图4.9 DesignService类verifyDesignSubmit方法代码

设计主管完成设计验证任务后，流程将会根据传入的参数自动流向下一步。

### 4.5.2 录入样衣版型

录入样衣版型的任务列表页面和设计验证一样，也是一个列表中显示每条任务的一些关键信息，点击右侧的详情按钮后，任务详细信息页面如图4.10所示。五个标签页中为订单的详情信息，设计主管在完成样衣的设计之后可以点击下方的选择文件按钮上传设计好的样衣版型cad文件，上传结束后点击提交按钮进行提交，文件将被上传到服务器保存。



图4.10 设计部模块录入样衣版型页面

录入样衣版型方法代码如图4.11所示。上传文件时，将文件的url地址、版本号、上传时间等信息保存到数据库中，同时将cad文件保存到服务器，并在以后的相关任务中提供历史版本的下载。

|  |
| --- |
| public boolean uploadDesignSubmit(int orderId, long taskId, String url,  Timestamp uploadTime) {  DesignCad designCad = null;  //查找cad文件信息，存在则更新信息，不存在则新建  List<DesignCad> designCadList = designCadDAO.findByOrderId(orderId);  if (designCadList.isEmpty()) {  designCad = new DesignCad();  designCad.setOrderId(orderId);  designCad.setCadVersion((short) 1);  } else {  designCad = designCadList.get(0);  short newVersion = (short) (designCad.getCadVersion() + 1);  designCad.setCadVersion(newVersion);  }  designCad.setCadUrl(url);  designCad.setUploadTime(uploadTime);  designCadDAO.attachDirty(designCad);  Map<String, Object> data = new HashMap<String, Object>();  try {  jbpmAPIUtil.completeTask(taskId, data, ACTOR\_DESIGN\_MANAGER);  return true;  } catch (InterruptedException e) {  e.printStackTrace();  return false;  }  } |

图4.11 DesignService类uploadDesignSubmit方法代码

录入样衣版型任务完成后，流程进入下一步。

### 4.5.3 设计生产验证

和录入样衣版型相比，设计生产验证多了一个cad文件的历史版本下载，其他部分和录入样衣版型类似，不再赘述。

### 4.5.4 生产版型确认

生产版型确认部分与设计生产验证类似，不在赘述。

## 4.6 生产部模块的实现

生产部模块主要包括生产验证、生产成本核算、样衣生产和批量生产等操作。本节描述了生产部模块各操作的具体实现细节，以及相关页面截图。

### 4.6.1 生产验证

生产验证的页面和代码和设计验证类似，只是操作者变成了生产主管，此处不再赘述。

### 4.6.2 生产成本核算

生产成本核算在进入任务详情后页面如图4.12所示，前五个标签页没有变化，都显示订单的详细信息，最后一个标签页为报价信息，生产主管将计算好的各种费用填入表格后，点击提交，后台会将输入的数据进行处理并同步到数据库。

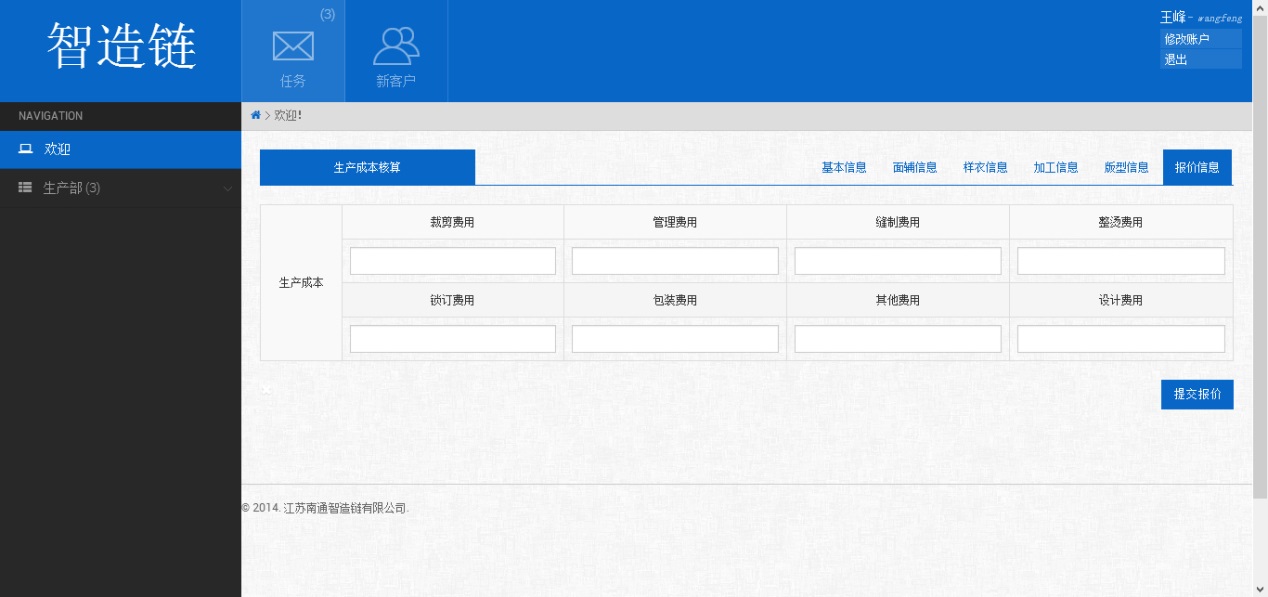


图4.12 生产部模块生产成本核算页面

生产成本核算的代码如图4.13所示。先查找报价单信息，如果已经存在则直接修改，没有的话则新建报价单，将报价信息以及计算好的总价全部填入后同步到数据库，最后完成任务。

|  |
| --- |
| public void computeProduceCostSubmit(int orderId, long taskId,  float cut\_cost, float manage\_cost, float nali\_cost,  float ironing\_cost, float swing\_cost, float package\_cost,  float other\_cost, float design\_cost) {  //通过订单id查找数据库中的报价信息，若没有则新建  Quote quote = QuoteDAO.findById(orderId);  if (quote == null) {  quote = new Quote();  quote.setOrderId(orderId);  ……//设置各种属性  QuoteDAO.save(quote);  } else {  //有的话则直接使用查找到的报价单  quote.setCutCost(cut\_cost);  ……//设置各种属性  }  //计算总价  float producecost = cut\_cost + manage\_cost + swing\_cost + ironing\_cost  + nali\_cost + package\_cost + other\_cost + design\_cost  + quote.getFabricCost() + quote.getAccessoryCost();  quote.setSingleCost(producecost);  QuoteDAO.attachDirty(quote);  Map<String, Object> data = new HashMap<String, Object>();  try {  //完成任务  jbpmAPIUtil.completeTask(taskId, data, ACTOR\_PRODUCE\_MANAGER);  } catch (InterruptedException e) {  e.printStackTrace();  }  } |

图4.13 ProduceService类computeProduceCostSubmit方法代码

生产成本核算任务完成后，流程进入下一步。

### 4.6.3 样衣生产

样衣生产的页面和代码和生产验证类似，在任务详情页面选择加工完成或者加工失败即完成任务。这里不再赘述。

### 4.6.4 批量生产

批量生产的任务详情页面如图4.14所示，前面六个标签页像是订单的详情信息，最后一个标签页为批量生产，批量生产任务结束后，生产主管在表格中填入实际生产的数量并选择加工是否完成，后台接收到数据后会进行处理并同步到数据库。



图4.14 生产部模块批量生产页面

批量生产的代码如图4.15所示。如果加工完成，则把从前端获得的加工单数据存入数据库中，并设置流程参数，完成任务。

|  |
| --- |
| public boolean pruduceSubmit(long taskId, boolean result,  List<Produce> produceList) {  if (result) {  for (int i = 0; i < produceList.size(); i++) {  produceDAO.save(produceList.get(i));  }  }  Map<String, Object> data = new HashMap<String, Object>();  try {  data.put(RESULT\_PRODUCE, result);  jbpmAPIUtil.completeTask(taskId, data, ACTOR\_PRODUCE\_MANAGER);  return true;  } catch (InterruptedException e) {  e.printStackTrace();  return false;  }  } |

图4.15 ProduceService类pruduceSubmit方法代码

完成批量生产任务后，流程进入下一步。

## 4.7 质检部模块的实现

质检部模块的任务为质量检查，与生产部模块的批量生产任务类似，都是填写加工单并提交，不再赘述。

## 4.8本章小结

本章主要内容为智造链系统设计部模块、生产部模块和质检部模块的详细设计和实现细节。详细设计部分介绍了三个模块的额详细类图、顺序图以及关键数据包设计，实现细节部分通过系统的运行页面截图以及部分方法代码描述了三个模块一些关键功能的实现细节，代码部分主要为验证、上传文件以及填写加工单的具体实现细节。

# 第五章 总结与展望

## 5.1 总结

本文首先在第一章里介绍了智造链系统的项目背景和工作流技术的发展历史，同时阐述了论文的主要工作和组织结构。

在第二章，主要介绍了系统中使用到的相关技术的概述，包括jBPM工作流框架、Spring框架、数据持久化工具Hibernate。jBPM框架是本系统能够比较容易的实现整个订单管理过程中的流程控制。

第三章对在需求和概要设计层面描述了本系统。需求分析包括系统用户角色、功能性需求、非功能性需求以及运行环境，同时介绍了设计部、生产部和质检部的用例描述、用例图和实体关系图。概要设计包括程序结构设计、功能组件划分以及接口设计。

第四章介绍了本系统的设计部模块、生产部模块和质检部模块的详细设计和部分关键功能的代码实现细节。详细介绍了设计部模块的设计验证和录入版型数据，以及生产部模块的批量生产。

在项目从需求分析到代码实现的整个过程中，本人运用了大学本科中学到的相关知识，积极参与，了解了软件开发的生命周期，对软件工程有了更加深入的理解。在项目开发过程中，本人通过实际的学习和使用，提升了对J2EE的理解，同时了解并学习了jBPM工作流的相关技术，最后将jBPM、Spring和Hibernate相结合，和组员共同完成了本项目的开发工作。

## 5.2 展望

目前，本文介绍的智造链系统的1.0版本基本完成，已经可以尝试在公司内部进行实际的使用。但是，在反复的测试以及使用过程中，我们发现系统仍然存在很大的提升空间。我们系统的页面还存在不足，在用户体验方面做的还不够好，有些地方会造成用户的不解，有些地方的操作还是太繁琐，从而导致用户的不满。我们会在今后的开发中继续改善这些地方。

本系统是以实际使用为目的的，因此，系统的后期维护和可扩展性是非常重要的。在后续的版本中，我们会加强这方面的开发和维护工作。本系统提供的是核心的业务流程以及一些简单的页面，具有良好的可扩展性，能根据公司自己的需要来改善系统，因此，本系统具有时分广阔的应用前景。

# 参考文献

[1] Nutt G J. The evolution towards flexible workflow systems[J]. Distributed Systems Engineering, 1996, 3(4): 276.

[2] Stoilova K P, Stoilov T A. Evolution of the workflow management systems[C]//Scientific Conference on Information, Communication and Energy Systems and Technologies-ICEST. 2006: 225-228.

[3] 罗海滨，范玉顺，吴澄，工作流技术总数，软件学报，2000.11（7）：899—907

[4] 龙志超，JBPM5工作流管理系统的改进及其在工程担保业务中的实现，硕士论文，湖南大学，2012

[5] Workflow Management Coalition，The Workflow Reference Model，Document Number TC00-1003，1995.1

[6] 范正顺，工作流管理技术基础，北京：清华大学出版社，2001

[7] 翟超杰，李兆瑞，jBPM工作流技术在绩效考核系统中的应用与改进，电脑编程技巧与维护， 2011年，18期，29—30

[8] 韩振文，JBoss Jbpm在财务信息系统中的应用研究[J],电脑与信息技术，2007,29（16），82-83

[9] 百度百科：Spring，http://baike.baidu.com/subview/23023/11192342.htm

[10] 孙卫琴，精通Hibernate: Java对象持久化技术详解，北京：电子工业出版社，2010，179-234页。

# 致谢

经过两个多月的项目开发和论文写作，本科毕业论文终于即将完稿。再次期间，我遇到了很多困难，但在老师和同学们的悉心帮助和鼓励下，这些困难最终都被解决了。

首先，感谢南京大学软件学院对我的培养，四年的本科生涯使我对软件工程有了系统的认识和足够的实践经验，这些都来自于软件学院提供的各种条件以及各位老师的指导。

我要感谢这两个多月来一直不断鼓励我、支持我的葛季栋老师。葛老师为人谦和，工作认真负责，对每一个学生都十分关心。在项目开发和论文写作过程中，葛老师的耐心指导给予了我很大的帮助。

另外，我要感谢和我一起奋斗两个多月的组员们，他们是莫其凡、王立铭和王健，还有从一开始就一直带着我们的研究生学长葛羽航，在项目开发期间，我遇到很多困惑和不解，通过请教学长和与组员们的讨论，不仅解决了问题，更对团队合作的重要性有了深刻的认识。

同时，我也要感谢辅导员黄蕾老师以及教务员王东霞老师，是她们无时不刻的关注着我们所有同学的毕业设计情况，及时通知我们最新的信息，在必要的时候给予我们帮助，是我们能够更好的完成毕业设计工作。

最后，我要感谢我的家人以及舍友。在他们的关怀和支持下，我顺利而充实地走过了本科四年的时光。

希望所有关心和帮助过我的人一生幸福、安康！