短期电力负荷预测方法

——灰色模型的应用

摘 要

短期负荷预测工作在电力市场运营中占据着越来越重要的地位，而短期电力负荷预测系统将负荷预测工作变得更加方便，将预测结果变得更加精确。该系统通过获取历史负荷数据，使用预测模型实现对电力的高精度预测，并且可以通过专业人员对预测结果进行修正从而使结果更加精确。本文按照软件开发的过程对短期电力负荷预测系统进行了全面分析，包括从一开始得到软件需求，经过需求分析，概要设计，详细设计，到最后代码实现与测试，完整地展现出设计软件的主要步骤。其中负荷预测模块是整个短期电力负荷预测系统的核心，采用的预测模型是灰色模型，并且实现了数据预处理，它也是连接其他功能模块的最重要的部分。数据管理模块是对历史负荷数据进行维护，其中包括录入24点负荷数据，查看负荷信息等。负荷分析起到决策支持的作用，帮助技术人员分析各个日期的负荷情况。系统管理模块实现了人员信息维护，权限分配，节假日信息维护，地区信息维护等功能。

**关键词**：电力负荷预测 灰色模型 软件系统

Short Term Electric Load Forecasting System

——Application of Grey Model

**ABSTRACT**

Short term load forecasting plays a more and more important role in the operation of power market, and short term power load forecasting system will be more convenient to load forecasting, and the result will be more accurate. By obtaining the historical load data, the system uses the prediction model to realize the high accuracy prediction of the power, and can be modified by the professional personnel to make the results more accurate. According to the software development process of short-term power load forecasting system conducted a comprehensive analysis, including from the beginning to get software requirements, through the demand analysis, outline design, detailed design to final code realization and test of complete shows the main steps of software design. The load forecasting module is the core of the short-term power load forecasting system, the prediction model adopted is grey model, and the realization of the data preprocessing. It is also connected to the most important part of other modules. Data management module is the maintenance of historical load data, including input 24 point load data, view the load information, etc.. Load analysis to play the role of decision support, to help technicians to analyze the load situation of each date. System management module to achieve the maintenance of personnel information, rights distribution, holiday information, regional information maintenance and other functions.

**Key Words:** Power Load Forecasting Grey Model Software System

目 录

[第一章 绪论 1](#_Toc450789754)

[1.1 项目背景 1](#_Toc450789755)

[1.2 国内外研究状况 1](#_Toc450789756)

[1.3 项目介绍 1](#_Toc450789757)

[第二章 系统实现技术简介 2](#_Toc450789758)

[2.1 JSP技术 2](#_Toc450789759)

[2.1.1 JSP简介 2](#_Toc450789760)

[2.1.2 JSP的优势 2](#_Toc450789761)

[2.2 JavaScript语言 2](#_Toc450789762)

[2.2.1 JavaScript概念及其特点 2](#_Toc450789763)

[2.2.2 JavaScript和Java的区别 3](#_Toc450789764)

[2.3 Struts2 3](#_Toc450789765)

[2.3.1 Struts2简介 3](#_Toc450789766)

[2.3.2 使用Struts2的原因 3](#_Toc450789767)

[2.4 B/S结构 3](#_Toc450789768)

[2.4.1 B/S结构简介 3](#_Toc450789769)

[2.4.2 B/S架构软件的优势与劣势 4](#_Toc450789770)

[第三章 需求分析 5](#_Toc450789771)

[3.1 业务分析 5](#_Toc450789772)

[3.1.1 业务流程图 5](#_Toc450789773)

[3.1.1 输入输出数据 6](#_Toc450789774)

[3.2 用例分析 6](#_Toc450789775)

[3.2.1 参与者 6](#_Toc450789776)

[3.2.2 用例图 6](#_Toc450789777)

[第四章 概要设计 11](#_Toc450789778)

[4.1 短期电力负荷预测系统模块划分 11](#_Toc450789779)

[4.1.1 功能模块图 11](#_Toc450789780)

[4.1.2 子模块设计 11](#_Toc450789781)

[4.2 系统架构设计 12](#_Toc450789782)

[4.3 数据库设计 13](#_Toc450789783)

[4.3.1 概念结构设计 14](#_Toc450789784)

[4.3.2 逻辑结构设计 15](#_Toc450789785)

[4.3.3 物理结构设计 16](#_Toc450789786)

[第五章 详细设计 19](#_Toc450789787)

[5.1 类图 19](#_Toc450789788)

[5.1.1 实体类图 19](#_Toc450789789)

[5.1.2 控制类图 19](#_Toc450789790)

[5.2 时序图 20](#_Toc450789791)

[5.2.1 数据维护时序图 20](#_Toc450789792)

[5.2.2 数据导入时序图 21](#_Toc450789793)

[5.2.3 负荷曲线时序图 22](#_Toc450789794)

[5.2.4 负荷特性时序图 22](#_Toc450789795)

[5.2.5 正常日预测时序图 24](#_Toc450789796)

[5.2.6 节假日预测时序图 24](#_Toc450789797)

[5.2.7 修正上报时序图 25](#_Toc450789798)

[5.2.8 账户维护时序图 26](#_Toc450789799)

[第六章 系统实现与代码编写 28](#_Toc450789800)

[6.1 系统实现 28](#_Toc450789801)

[6.1.1 用户登录 28](#_Toc450789802)

[6.1.2 数据管理 29](#_Toc450789803)

[6.1.3 负荷分析 31](#_Toc450789804)

[6.1.4 负荷预测 33](#_Toc450789805)

[6.1.5 系统设置 36](#_Toc450789806)

[第七章 测试 38](#_Toc450789808)

[7.1 测试的目的 38](#_Toc450789809)

[7.2 测试过程 38](#_Toc450789810)

[7.2.1 数据管理测试 38](#_Toc450789811)

[7.2.2 负荷分析测试 40](#_Toc450789812)

[7.2.3 负荷预测测试 41](#_Toc450789813)

[7.2.4 系统管理测试 43](#_Toc450789814)

[7.3 测试结论 4](#_Toc450789814)4

[第八章 总结 46](#_Toc450789815)

[8.1 结论 46](#_Toc450789816)

[8.2 展望 46](#_Toc450789817)

参考文献

附 录

致 谢

# 绪论

## 项目背景

电力负荷预测工作是根据电力负荷的发展变化规律，预计或判断其未来负荷发展趋势和状况的活动。短期电力负荷预测是电力系统调度、用电、计划、规划的重要依据。提高负荷预测技术水平，有利于计划用电管理，有利于合理安排电网运行方式和机组检修计划，保证社会的正常生产和生活，有效地降低发电成本。利用短期负荷预测的结果来安排和制定生产计划，可以减少不必要的备用容量，提高电力系统经济效益和社会效益。同时，负荷预测精度的提高，对于经济优化地制订发电计划、制订经济合理的电力调配计划，制订上网竞价计划、在竞价上网中取得优势、控制电网经济运营、保障生产和生活用电等方面具有重要意义。因此，电力系统负荷预测研究已成为电力工程界研究的一个重要领域[1-2]。

## 国内外研究状况

短期负荷预测工作将在电力市场运营中占据十分重要的地位。国内外的许多专家、学者在预测理论和方法方面做了大量的研究工作，取得了很多卓有成效的进展。目前用于短期负荷预测的方法很多，可以分为经典预测方法，传统预测方法和智能预测方法等。国内市场也有类似的软件，很多是嵌入EMS系统当中的，采用C/S模式，以SqlServer作为后台数据库，选用面向对象的可视化的C++ Builder开发核心程序和系统界面。这些软件不仅实现了对电力的高精度预测，而且还有专家系统进行修正提高精确，使用效果良好。国外有MATLAB等软件，MATLAB具有数值计算，数据可视化功能和易于使用的编程开发环境，这个软件可以进行仿真分析从而预测短期电力，在电力预测理论分析方面得到运用。

## 1.3 项目介绍

电力负荷系统是典型的灰色系统，因而灰色预测技术在电力系统中有着广泛的应用，其短期预测精度高。利用灰色模型(Grey Model)的GM(1，1)及其改进模型，本项目成功地实现了对普通日电力负荷预测与特殊日电力负荷预测。同时，该项目还能对电力负荷进行分析，提供了决策支持，而且具有高度的扩展性。实现了网络化数据管理与科学计算的高度一体化，为电力系统各部门进行详尽的数据分析和高质量的需求预测提供了灵活的操作平台[3]。

# 系统实现技术简介

## 2.1 JSP技术

### 2.1.1 JSP简介

JSP(Java Server Pages)是由Sun Microsystems公司倡导、许多公司参与一起建立的一种动态网页技术标准。 JSP技术是用JAVA语言作为脚本语言的，JSP网页为整个服务器端的JAVA库单元提供了一个接口来服务于HTTP的应用程序。利用这一技术，可以建立先进、安全和跨平台的动态网站[4]。

在传统的网页HTML文件(\*.htm,\*.html)中加入Java程序片段(Scriptlet)和JSP标记(tag)，就构成了JSP网页(\*.jsp)。Web服务器在遇到访问JSP网页的请求时，首先执行其中的程序片段，然后将执行结果以HTML格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送email等等，这就是建立动态网站所需要的功能。所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求最低，可以实现无Plugin，无ActiveX，无Java Applet，甚至无Frame。

### 2.1.2 JSP的优势

对于用户界面的更新，其实就是由 Web Server进行的，所以给人的感觉更新很快。所有的应用都是基于服务器的，所以它们可以时刻保持最新版本。客户端的接口不是很繁琐，对于各种应用易于部署、维护和修改。

## 2.2 JavaScript语言

### 2.2.1 JavaScript概念及其特点

JavaScript是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言。使用它的目的是与HTML超文本标记语言、Java 脚本语言（Java小程序）一起实现在一个Web页面中连接多个对象，与Web客户交互作用[5]。它是通过嵌入或调入到标准的HTML语言中实现的,具有以下几个基本特点：

1、是一种脚本编写语言 JavaScript是一种脚本语言，它采用小程序段的方式实现编程。像其它脚本语言一样,JavaScript同样也是一种解释性语言。

2、动态性的 JavaScript是动态的，它可以直接对用户或客户输入做出响应，无须经过Web服务程序。

3、跨平台性 JavaScript是依赖于浏览器本身，与操作环境无关，只要能运行浏览器的计算机，并支持JavaScript的浏览器就可正确执行。

### 2.2.2 [JavaScript和Java的区别](http://edu.100down.com/it/web/JavaScript/151022396.html)

JavaScript 创作者可以不那么注重程式技巧，所以许多 Java 的特性在 JavaScript 中并不支援,同时两者却是两个公司开发的不同的两个产品。Java是SUN公司推出的新一代面向对象的程序设计语言，特别适合于Internet应用程序开发；而JavaScript是Netscape公司的产品，其目的是为了扩展Netscape Navigator功能,而开发的一种可以嵌入Web页面中的基于对象和事件驱动的解释性语言, 它的前身是Live Script；而Java的前身是Oak语言。

## 2.3 Struts2

### 2.3.1 Struts2简介

Struts2是一个非常流行的具有高度可扩展性的MVC框架，它整合了两个优秀的MVC框架：传统的Struts和WebWork[6]。Struts2的体系与Struts1体系的差别非常大，因为Struts2使用了WebWork的设计核心，而不是Struts1的设计核心。Struts2中大量使用拦截器来处理用户请求，从而允许用户的业务逻辑器与Servlet API分离。

### 2.3.2 使用Struts2的原因

1.它是建立在MVC这种公认的好的模式上的，Struts在M、V和C上都有涉及，但它主要是提供一个好的控制器和一套定制的标签库上，它的着力点在C和V上，因此，它天生就有MVC所带来的一系列优点，如：结构层次分明，高可重用性，增加了程序的健壮性和可伸缩性，便于开发与设计分工，提供集中统一的权限控制、校验、国际化、日志等等；

2.其次，它是个开源项目，得到了包括它的发明者Craig R.McClanahan在内的一些程序大师和高手持续而细心的呵护，并且经受了实战的检验，使其功能越来越强大，体系也日臻完善；

3.它对其他技术和框架显示出很好的融合性。

## 2.4 B/S结构

### 2.4.1 B/S结构简介

B/S（Browser/Server）结构即浏览器和服务器结构。它是随着Internet技术的兴起，对C/S结构的一种变化或者改进的结构[7]。在这种结构下，用户工作界面是通过WWW浏览器来实现，极少部分事务逻辑在前端（Browser）实现，但是主要事务逻辑在服务器端（Server）实现，形成所谓三层3-tier结构。这样就大大简化了客户端电脑载荷，减轻了系统维护与升级的成本和工作量，降低了用户的总体成本（TCO）。

### 2.4.2 B/S架构软件的优势与劣势

1．维护和升级方式简单。目前，软件系统的改进和升级越来越频繁，B/S架构的产品明显体现着更为方便的特性。

2．成本降低，选择更多。凡使用B/S架构的应用管理软件，只需安装在Linux服务器上即可，而且安全性高。Linux除了操作系统是免费的以外，连数据库也是免费的，这种选择非常盛行。

3．应用服务器运行数据负荷较重，一旦发生服务器“崩溃”等问题，后果不堪设想。因此，许多单位都备有数据库存储服务器，以防万一。

# 需求分析

需求分析是通过分析业务，得到系统所需要完成的功能，确定输入以及输出的内容。本章通过对电力预测业务分析，得到一些系统的宏观目标，为以后的一步步分析奠定基础。

## 3.1 业务分析

### 3.1.1 业务流程图

通过对电力公司业务的全面分析，得到如图3.1所示的活动图，该活动图对有关电力负荷预测的业务进行了详尽地描述。

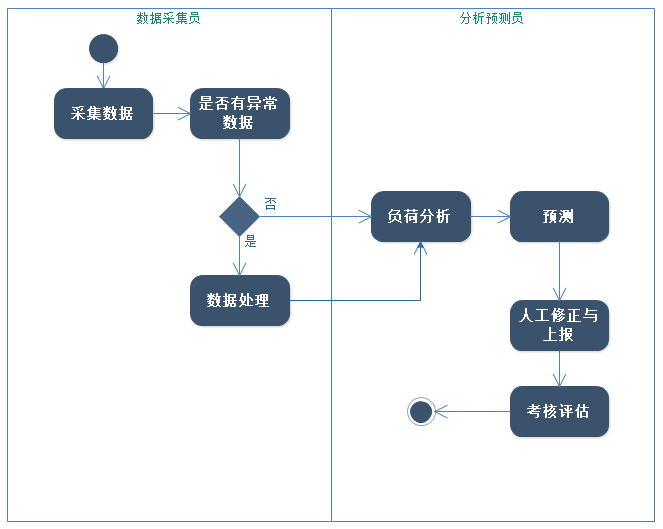


图3.1 业务流程图

Fig.3.1 Business Process diagrams

首先由数据采集员采集历史电力负荷相关数据，包括负荷值，日期以及气象相关因素等数据，然后再对这些数据进行预处理，排查异常数据。数据收集好以后，交给分析预测员，分析预测员会对现有数据进行分析总结，采取比较好的预测模型进行预测，然后修正不合理的预测数据，最后进行上报。相关领导通过对分析的数据进行总结后制订相应计划。

### 3.1.1 输入输出数据

**（1）输入数据：**内部已有的数据：节假日数据、地区数据、预测模型相关数据、员工数据；手工录入的数据：电力负荷数据、气象相关数据（包括气候，最高最低温度等）、日期。

**（2）输出数据：**24点预测负荷数据、负荷曲线、日负荷特性、相关报表等。

## 3.2 用例分析

### 3.2.1 参与者

在负荷数据录入模块，参与者主要是数据采集员，在负荷预测模块，参与者主要是分析预测员，对于系统全局的设置，人员的管理，参与者主要是系统管理员。

### 3.2.2 用例图

涉及到数据采集员的用例图如图3.2所示。

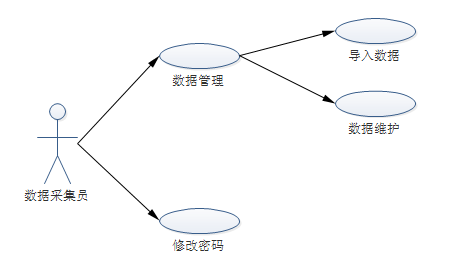


图3.2 数据采集员用例图

Fig.3.2 Data collector use case diagram

关键用例描述如表3.1与表3.2所示：

表3.1 数据管理用例描述

Table3.1 Data management use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 数据管理 |
| 用例描述 | 采集员导入历史负荷数据，并且可以进行查询以及修改删除 |
| 参与者 | 数据采集员 |
| 前置条件 | 收集到了相关数据，身份验证合格 |
| 后置条件 | 如果这个用例成功，在系统中建立历史负荷数据记录 |
| 基本操作流程 | 1, 数据采集员输入负荷数据  2，系统验证数据合法性  3，数据采集员处理异常数据  4，数据维护完毕 |
| 可选操作流程 | 1. 取消相关操作 2. 数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 导入数据，数据维护 |
| 被拓展的用例 | 无 |

表3.2 修改密码用例描述

Table3.2 Modify password use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 修改密码 |
| 用例描述 | 采集员修改自己登陆系统的密码 |
| 参与者 | 数据采集员 |
| 前置条件 | 进入系统 |
| 后置条件 | 如果这个用例成功，登入系统密码改变 |
| 基本操作流程 | 1，数据采集员输入密码  2，数据采集员输入确认密码  3，系统进行合法性验证  4，修改成功 |
| 可选操作流程 | 1，取消相关操作  2，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 无 |
| 被拓展的用例 | 无 |

涉及到分析预测员的用例图如图3.3所示。

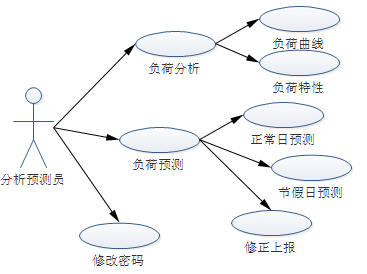


图3.3 分析预测员用例图

Fig.3.3 Analysis and forecast of the use case diagram

关键用例描述如表3.3与表3.4所示：

表3.3 负荷分析用例描述

Table3.3 Load analysis use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 负荷分析 |
| 用例描述 | 分析预测员可以查看现有负荷数据的各种状态 |
| 参与者 | 分析预测员 |
| 前置条件 | 存在相关数据，身份验证合格 |
| 后置条件 | 如果这个用例成功，将得到汇总分析后的负荷数据 |
| 基本操作流程 | 1，分析预测员输入负荷分析方式  2，分析预测员输入查询条件  3，根据相应的结果制订计划 |
| 可选操作流程 | 1. 取消相关操作   2，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 负荷曲线，负荷特性 |
| 被拓展的用例 | 无 |

表3.4 负荷预测用例描述

Table3.4 Load forecasting use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 负荷预测 |
| 用例描述 | 通过现有历史数据对未来的电力负荷进行预测 |
| 参与者 | 分析预测员 |
| 前置条件 | 存在相关历史数据，身份验证合格 |
| 后置条件 | 如果这个用例成功，将输出某日的预测负荷数据 |
| 基本操作流程 | 1，分析预测员选择预测方式   1. 分析预测员输入预测日条件   3，系统进行数据预处理后产生预测结果  4，分析预测员对预测结构手动修正并且上报 |
| 可选操作流程 | 1，取消相关操作  2，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 正常日预测，节假日预测，修正上报 |
| 被拓展的用例 | 无 |

涉及到系统管理员的用例图如图3.4所示。

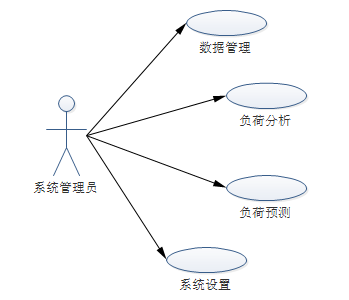


图3.4 系统管理员用例图

Fig.3.4 System administrator use case diagram

系统管理员除了有分析预测员和数据收集员的用例以外，还有如表3.5所示的用例：

表3.5 系统设置用例描述

Table3.5 System settings use case description

|  |  |
| --- | --- |
| **描述项** | **说明** |
| 用例名称 | 系统设置 |
| 用例描述 | 系统管理员可以设置系统的一些全局特性，比如预测节假日选取，登录人员的维护等等 |
| 参与者 | 系统管理员 |
| 前置条件 | 存在相关数据，身份验证合格 |
| 后置条件 | 如果这个用例成功，将修改系统的一些设置，比如预测所需要选择的节假日，预测方法的选取 |
| 基本操作流程 | 1，系统管理员收到相关变更要求  2，对节假日，用户，预测方法按照目标进行修改 |
| 可选操作流程 | 1. 取消相关操作   4，数据输入不合法，操作失败 |
| 被泛化的用例 | 无 |
| 被包含的用例 | 无 |
| 被拓展的用例 | 无 |

# 第四章 概要设计

概要设计的主要任务是把[需求分析](http://baike.baidu.com/view/111493.htm)得到的系统模型转换为[软件结构](http://baike.baidu.com/view/600142.htm)和[数据结构](http://baike.baidu.com/view/9900.htm)。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的[层次结构](http://baike.baidu.com/view/420833.htm)及调用关系、确定模块间的接口及人机界面等。数据[结构设计](http://baike.baidu.com/view/411272.htm)包括数据特征的描述、确定数据的结构特性、以及[数据库](http://baike.baidu.com/view/1088.htm)的设计。

## 4.1 短期电力负荷预测系统模块划分

### 4.1.1 功能模块图

基于以上需求，通过对需求进行整合与整理，短期电力负荷预测系统的总功能模块划分如图4.1所示：

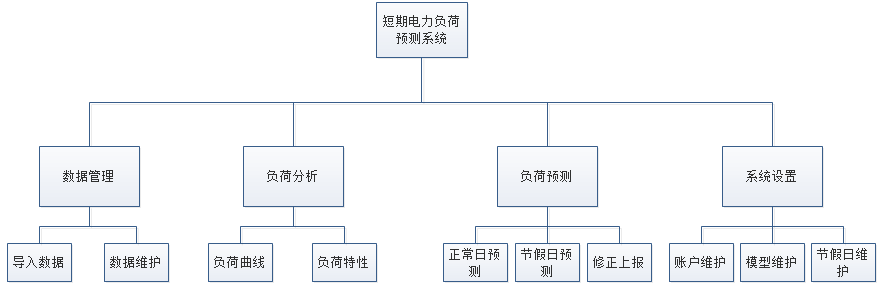


图4.1 短期电力负荷预测系统功能模块

Fig.4.1 Function module of short term power load forecasting system

### 4.1.2 子模块设计

数据管理子模块：能够录入不同时间点的用电负荷以及此时的气象指数，从而实现数据的录入，为数据预处理做好准备，同时支持多种录入方式。操作员可以也可以对录入的历史数据进行查询，删除，修改，排除异常数据。

负荷分析子模块：可以查看历史日负荷的特性，包括日最大负荷、日最小负荷、日峰谷差、日峰谷差率日电量、日平均负荷、日负荷率、日最小负荷率等。还可以查看负荷的罗列曲线，并且可以对不同日期的负荷曲线进行对比。

负荷预测子模块：这是整个系统中最重要的一部分，预测方法的选择很大程度上影响负荷预测的精度。该功能根据历史数据，进行数据挖掘，根据GM(1,1)模型进行第二天的数据预测。

系统设置子模块：只有系统管理员能操作这部分功能，包括账户维护，模型维护，节假日维护。账户维护是对用户的综合管理，包括设置用户权限，增加，修改，删除，查看用户；模型维护主要是对负荷预测方法的动态扩展，使系统更具扩展性；节假日维护是用来设置负荷预测所需要用到的节假日数据。

## 4.2 系统架构设计

本系统采用B/S模式，选择SQL Server 2000作为后台数据库，选择JAVA、JSP、JavaScript、Html作为应用程序开发工具，运用Tomcat服务器技术，整个系统完全基于Browser/Server模式进行设计。

系统的各个包的引用关系如图4.2所示。该图体现出系统架构设计的MVC思想，其中JSP作为View层，提供对数据的显示；包com.yuan.action与包com.yuan.service共同构成了Controller层，负责对数据的控制以及视图的显示，包com.yuan.model构成了Model层，是搭载数据的载体。包com.yuan.util是一个工具包，用于存放各种工具类，比如统一访问数据库的类，访问XML文件的类。

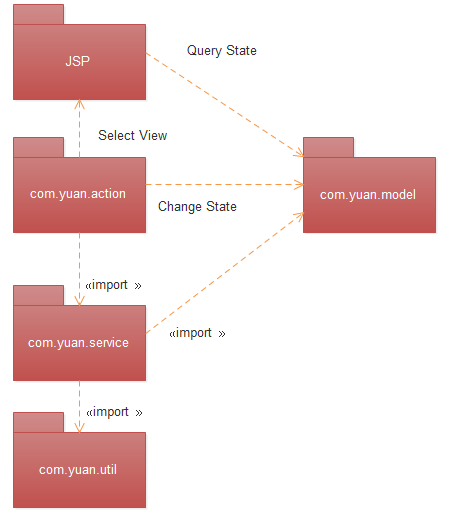


图4.2 包引用图

Fig.4.2 Package reference diagram

## 4.3 数据库设计

数据库设计是建立数据库及其应用系统的关键技术，是信息系统开发和建设的核心技术，具体说，数据库设计是指对于一个给定的应用环境，构造最优的数据模式，建立数据库及其应用系统，使之能够有效地存储数据，满足各种用户的应用需求（信息要求和处理要求）。

按照规范设计的方法，考虑数据库及其应用系统开发全过程，将数据库设计分为如图4.3所示六个阶段。

需求收集和分析

设计概念结构

设计逻辑结构

数据模型优化

设计物理结构

评价设计，性能预测

物理实现

试验性运行

使用、维护数据库

应用需求

（数据、处理）

转换规则

DBMS功能

优化方法

应用要求

DBMS详细特征

不满意

不满意

需求分

析阶段

概念设

计阶段

逻辑设

计阶段

物理设

计阶段

数据库

实施阶段

数据库运行

维护阶段

图4.3 数据库设计步骤

Fig.4.3 Database design steps

### 

### 4.3.1 概念结构设计

将需求分析得到的用户需求抽象为信息结构即概念模型的过程就是概念结构设计。它是整个数据库设计的关键。

描述概念模型的有力工具是E-R（Entity-Relationship，实体-联系）模型，本系统就是采用这种工具进行的概念模型设计。

设计概念结构通常有四种方法：自顶向下、自底向上、逐步扩张和混合策略（即将自顶向下和自底向上相结合）。本系统采用的是自底向上的方法，即自顶向下进行需求分析，然后再自底向上的设计概念结构。

在对短期电力负荷预测系统进行分析的时候，由之前的需求分析可以得出如图4.4所示ER图：

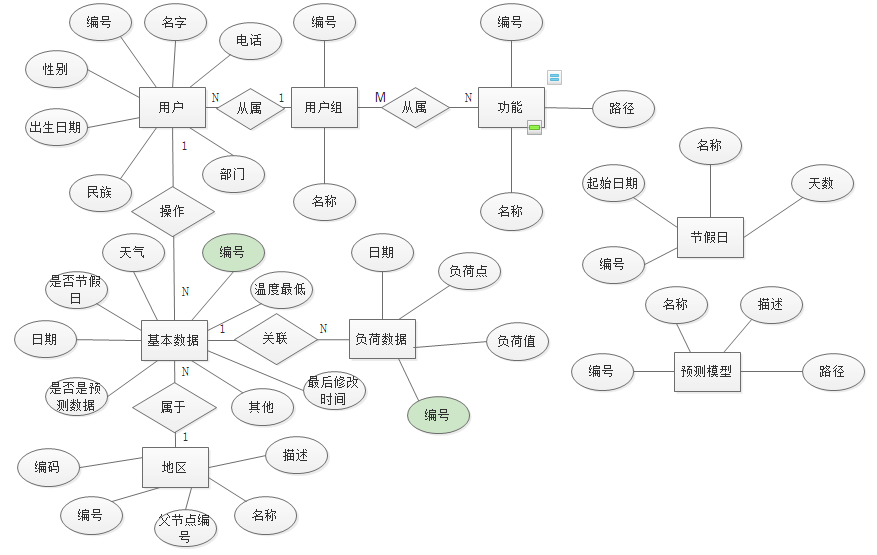


图4.4 短期电力负荷预测系统ER图

Fig.4.4 Short term electric load forecasting system ER diagram

各个实体与实体联系是依靠数据流图与实际情况得来，具体如下：

为了方便用户权限控制，系统可以允许为用户分配用户组，一个用户会被分配在一个用户组当中，一个用户组下可能有多个用户，所以用户与用户组的关系是一对多的关系。每个用户组可以拥有多个权限功能，而一个功能同样也可以属于多个用户组，所以用户组与功能是多对多的关系。

一个用户可以操作多条基本数据，而一条基本数据的最后一次操作只有一人，所以用户与基本数据的关系是一对多的关系。

一条基本数据记录的是某一地区某天的电力使用情况，而一个地区可以记录多次负荷基本数据，所以基本数据与地区的关系是多对一的关系。

一条基本数据是某天负荷记录的基本情况，而一天当中需要记录24点负荷，所以基本数据与负荷数据之间是一对多的关系。

节假日与预测模型没有和其他实体产生联系。

### 4.3.2 逻辑结构设计

由ER图可以得到如下的关系模式以及范式规范说明：

1.用户（编号，名字，电话，性别，出生日期，民族，部门，用户组编号）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以用户属于BCNF

2.用户组（编号，名称，功能编号）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以用户组属于BCNF

3.功能（编号，名称，路径）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

4.基本数据（编号，天气，是否节假日，日期，是否是预测数据，最高温度，最低温度，最后修改时间，其他，地区编号）

主键：编号

函数依赖{编号->其他所有属性，（日期，地区编号）->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

5.地区（编号，编码，父节点编号，名称，描述）

主键：编号

函数依赖{编号->其他所有属性，编码->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

6.负荷数据（编号，负荷点，负荷值，基本数据编号）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

7.节假日（编号，名称，起始日期，天数）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

8.预测模型（编号，名称，路径，描述）

主键：编号

函数依赖{编号->其他所有属性}

由函数依赖可知，依赖左端为候选码，所以功能属于BCNF

### 4.3.3 物理结构设计

为给定的逻辑数据模型选取一个最合适应用要求的物理结构的过程就是数据库的物理设计。物理设计时要确定数据库的存储路径，数据规模等，在数据库管理系统中创建数据库，建立数据库的所有数据模式。

本系统数据库在设计过程中参考E-R模型设计，充分考虑了数据表的划分及其之间的联系，在保障数据流向简明、查询效率较高的前提下，设计了如表4.1~表4.8所示的数据库表结构：

1，负荷基本数据表（BasicData）

表4.1 负荷基本数据表

Table4.1 Load basic data table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 记录编号（主键） | Int | Not null |
| electricDate | 负荷日期 | date | null |
| isHoliday | 是否节假日 | Int | null |
| highTemp | 最高温度 | float | null |
| lowTemp | 最低温度 | float | null |
| weather | 气候 | nvarchar(50) | null |
| isForecast | 是否是预测数据 | int | null |
| updateDate | 修改日期 | datetime | null |
| updateUserId | 修改人员（外键） | Int | null |
| other | 备注 | nvarchar(50) | null |
| areaId | 地区编号（外键） | int | null |

说明：这一张表存放历史数据与预测数据两种数据，用isForecast字段来进行区分，气候分为三种：晴、阴、雨。

2，负荷值表（LoadValue）

表4.2 负荷值表

Table4.2 Load value table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 记录编号（主键） | Int | Not null |
| loadTime | 负荷点 | int | null |
| value | 负荷值 | float | null |
| basicId | 基本数据Id（外键） | int | null |

说明：负荷点字段存放的是24点负荷的负荷点，取值范围是1-24之间的整数。

3，用户表（t\_User）

表4.3 用户表

Table4.3 User table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 用户ID（主键） | Int | Not null |
| loginName | 用户名 | nvarchar(50) | null |
| password | 密码 | nvarchar(50) | null |
| name | 姓名 | nvarchar(50) | null |
| tel | 电话 | nvarchar(50) | null |
| gender | 性别 | nvarchar(50) | null |
| department | 部门 | nvarchar(50) | null |
| birthday | 生日 | date | null |
| nation | 民族 | nvarchar(50) | null |
| groupId | 用户组Id（外键） | int | null |

说明：用户组字段是用来将用户和所属的角色对应起来，实现权限分配的功能，用户使用用户名登录系统。

4，用户组表（UserGroup）

表4.4 用户组表

Table4.4 User group table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 用户组ID（主键） | Int | Not null |
| name | 用户组名称 | nvarchar(50) | null |

说明：系统分为三个用户组：系统管理员、数据采集员、预测分析员。

5，功能表（t\_Function）

表4.5 功能表

Table4.5 Function table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 功能ID（主键） | Int | Not null |
| name | 功能名称 | nvarchar(50) | null |
| path | 网页路径 | nvarchar(50) | null |
| pid | 父功能Id | int | null |
| pic | 图片 | nvarchar(50) | null |

说明：这个表通过自反关联，实现了树状结构功能展示，其中pid是外键，对应该表的主键；每个功能展示出来的时候对应有一张图片表示该功能，这张表的pic就是用来存储功能图片的。

6，功能用户组联系表（G\_F\_Relation）

表4.6 功能用户联系表

Table4.6 The table of relation about function and user

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 功能ID（主键） | Int | Not null |
| groupId | 用户组ID（外键） | int | null |
| functionId | 功能ID（外键） | int | null |

说明：这个表通过俩个外键将用户组表与功能表联系起来，实现了一个用户

拥有多个权限的功能。

7，节假日表（Holiday）

表4.7 节假日表

Table4.7 Holiday table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 节假日ID（主键） | Int | Not null |
| name | 节假日名称 | nvarchar(50) | null |
| start | 节假日起始日期 | date | null |
| interval | 假期天数 | int | null |

说明：主要为节假日预测提供相应的节假日数据。

8，地区表（Area）

表4.8 地区表

Table4.8 Area table

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **字段解释** | **类型** | **是否可空** |
| Id | 地区ID（主键） | Int | Not null |
| code | 地区编码 | nvarchar(50) | null |
| pid | 父节点 | int | null |
| name | 地区名 | nvarchar(50) | null |
| description | 地区描述 | nvarchar(50) | null |

说明：这张表通过自反关联实现了多级地区的查询，其中pid是外键。

# 第五章 详细设计

详细[设计](http://baike.baidu.com/view/14417.htm)是[软件工程](http://baike.baidu.com/view/1659.htm)中[软件开发](http://baike.baidu.com/view/190611.htm)的一个步骤，就是对[概要设计](http://baike.baidu.com/view/551728.htm)的一个细化，就是具体地设计每个模块实现算法以及所需的[局部](http://baike.baidu.com/view/470791.htm)结构。利用灰色模型(Grey Model)的GM(1，1)及其改进模型，本文成功地进行了普通日电力负荷预测与特殊日电力负荷预测。

5.1 GM(1，1)模型的建立方法[15~17]

1.对随机序列作一次累加(1…AGO)生成序列其中.（式5.1）

2.按照Xi(1)的指数增长规律，可知Xi(1)满足下列一阶线性微分方程：

（式5.2）

(X(1)是时间t的函数，这是灰色方程，部分数据未知)

3.参数估计：记待定。经离散化处理，得：Yn=BA.使用最小二乘法求出A的近似解：

（式5.3）

将近似值,代入原微分方程：

（式5.4）

(\*)(原微分方程的白化方程)

其中

（式5.5）

4.Xi(1)的预测值：求解微分方程(\*)得到原微分方程的近似解

（式5.6）

其中X1(1)=X1(0)；写成离散形式，得到Xi(1)的预测值：

（式5.7）

5.Xi(0)的预测值：

（式5.8）

对呈良好增长趋势的变化过程，用GM(1，1)都能得到较好的精确度，但有时遇到的变化过程较差的增长趋势，用一次GM(1，1)得不到满意的精确度，此时为了得到更好的精确度，常对其残差序列再进行一次GM(1，1)拟合，这就是常说的GM(1，1)改进模型。模型的精确度可通过已知的前n个历史数据与其相应的n个预测数据比较，若精确度较好，则直接预测下一个未知数据。否则，要进行修正。

5.2 普通日电力负荷预测

5.2.1 灰色差值预测模型

GM(1，1)模型不需要历史数据具有明显的趋势，所需历史数据较少，因此适用范围广泛。但GM(1，1)模型不能反映电力负荷周期性的特性，而普通日的短期电力负荷却具有明显的周期性，这种周期性呈现明显的日周期性和周周期性。所谓日周期性，指每日的同一时刻(24点)，用电量基本相同。所谓周周期性，指每周的同一天同一时刻(24点)，用电量基本相同。

为此，本文使用一种差值灰色模型来预测普通日的短期电力负荷，收到较好效果。差值灰色预测模型是用相邻的两天同一时刻的负荷数据构成差值，以周为间隔且按时间先后顺序构成的差值列作为灰色建模历史数据列。为预测未来某点(24点)负荷(如星期三8点)，从历史数据中选取过去的最近的几个星期三8点的电力负荷数据构成数据列{y2(t)｜t=1,2,3…,n}(n个星期)(为方便，称之为“当天数列”)，y2(n+1)为本星期三8点的预测负荷。相应星期二8点的电力负荷数据构成数列{y1(t)｜t=1,2,3…,n}(为方便，称之为“昨天数列”)。令

x(0)(t)=y2(t)-y1(t),t=1,2,3，…,n, （式5.9）

则得差值序列

x(0)={x(0)(t)｜t=1,2,3，…,n}.（式5.10）

其中y2(t)｜t=1,2,3,…，n及y1(t)｜t=1,2,3,…，n,n+1为已知，y2(n+1)的预测值为

.则该差值序列的周期性已被消除，得到一个随机序列。由于差值序列可能有负项，则取

y(0)={y(0)(t)｜t=1,2,3,…,n}={｜x(0)(t)｜t=1,2,3,…,n}，（式5.11）

建立该序列的GM(1，1)模型，同上得到其预测值：

(只取第一个预测值)。（式5.12）

符号修正：

，（式5.13）

其中

，

得到本周三8点的预测负荷。同理，可得本周三一天24点的预测负荷。

这种差值灰色模型是对GM(1，1)模型的一种改进，该方法充分利用了最近的历史数据且充分考虑了电力负荷的周期性，适用于普通日电力负荷预测。

* + 1. 数据预处理

为了提高预测的准确度，对于用差值灰色预测模型使用的实际数据先进行数据预处理。所谓数据预处理，简而言之就是剔除异常数据。本文采用滑动平均法进行数据预处理，效果理想。

记原始数列为x0 ,对于两端点，可用以下计算公式处理：

，（式5.14）

，（式5.15）

对于中间各数据点，采用下式计算其滑动平均值：

，（式5.16）

* + 1. 应用效果

经试算，普通日负荷预测准确度满足用户要求，最大相对误差在5%以内。下面附某日电网的负荷预测日报表(表1).

同样的预测模型有时预测结果精确度不同，关键之处就在于数据的预处理。在该模型的试算阶段未将周末、节假日、气候温度急变这些数据(全天24点)作为异常数据予以剔除，而是直接用于确定差值灰色预测模型的各种待定参数，以此模型来预测日负荷，预测准确度大部分满足不了用户提出的大于95%的要求。如果数列中的数据含有周末、节假日、气候温度剧变等日期的数据，那么，用得到的差值灰色模型来预测当天的负荷时，预测准确度就不一定满足用户要求。可见，对某一变化过程进行预测时，数据预处理的方法与预测模型同等重要。

表5.1 负荷预测日报表

Table 5.1 Load forecasting statements Japan

|  |  |  |  |
| --- | --- | --- | --- |
| 时 间 | 预测负荷 | 实际负荷 | 相对误差（%） |
| 01:00 | 1865 | 1833 | 1.71 |
| 02:00 | 1778 | 1762 | 0.90 |
| 03:00 | 1783 | 1753 | 1.68 |
| 04:00 | 1760 | 1728 | 1.82 |
| 05:00 | 1780 | 1763 | 0.96 |
| 06:00 | 1861 | 1885 | 1.29 |
| 07:00 | 2008 | 2106 | 4.88 |
| 08:00 | 2137 | 2140 | 0.14 |
| 09:00 | 2310 | 2345 | 1.52 |
| 10:00 | 2350 | 2370 | 0.85 |
| 11:00 | 2470 | 2464 | 0.02 |
| 12:00 | 2390 | 2332 | 2.43 |
| 13:00 | 2163 | 2116 | 2.17 |
| 14:00 | 2181 | 2160 | 0.96 |
| 15:00 | 2257 | 2294 | 1．64 |
| 16:00 | 2290 | 2262 | 1.22 |
| 17:00 | 2511 | 2439 | 2.87 |
| 18:00 | 2582 | 2555 | 1.05 |
| 19:00 | 2601 | 2537 | 2.46 |
| 20:00 | 2986 | 2949 | 1.24 |
| 21:00 | 2902 | 2825 | 2.65 |
| 22:00 | 2603 | 2567 | 1.38 |
| 23:00 | 2317 | 2231 | 3.71 |
| 24:00 | 1983 | 1916 | 3.38 |
| 日预测平均准确率(%) 98.21  日最高负荷计：2986  日最低负荷计：1760 | | 实际：2949  实际：1728 | 准确率(%)：98.76  准确率(%)：98.18 |

5.3 特殊日电力负荷预测

对周末、节假日这类变化过程的日负荷值，仍呈24小时周期性规律变化，仅因某些人为的因素而使其负荷值的波动幅度超过日常平衡状态下的负荷值，因而导致用适合普通日负荷预测的灰色差值模型或ARIMA(p,d,q)模型来预测周末、节日这类特殊变化过程的日负荷值准确率较差。对差值进行纵向(不同年的相同采样点)比较发现：各年同点差值绝大部分呈较好的增长规律变化，因而，我们决定采用ARIMA(p,d,q)模型与GM(1，1)改进模型对特殊日电力负荷进行组合预测。用适合普通日负荷预测的ARIMA(p,d,q)模型来预测周末、节假日这类特殊变化过程的日负荷基值，并为GM(1，1)改进模型构造残差序列，对残差序列进行GM(1，1)拟合，以此来提高对周末、节假日这类特殊变化过程的日负荷预测值的准确率。

5.3.1 GM(1，1)改进模型的建立方法

1.对序列Xt(0)，t=1,2,…,n,按GM(1，1)模型方法求出预测值,t=1,2,…，n,算出残差序列εt(0)，t=1,2,…,n,其中

.（式5.17）

2.若k≥k0时，εk(0)符号一致(预测结果较差)，

k=k0，k0+1，k0+2，…，n，（式5.18）

则对残差子序列利用GM(1，1)中的方法求出预测值，，其中

(残差模型)。（式5.19）

3. 的预测修正值：

,（式5.20）

其中“±”的取定应与残差εk(0)(k≥k0)的符号一致。通过残差模型，得到新的预测残差。经过和序列Xt(0) t=1,2,…，n新的预测值的计算，一般可获取较高的精度。

5.3.2 特殊日负荷残差序列的建立和预测方法

我们为GM(1，1)改进模型构造差值序列(i代表某年、或某月、或某日，t代表某时)，对某一确定的t，将代入GM(1，1)改进模型中得到残差值预测值，再得到的预测修正值：.经试算，准确度明显提高，绝大部分周末与节假日的日负荷预测准确率满足用户要求，大于95%。具体方法如下：

1.对序列Zit,i=1,2,…n,t=1,2,…24用ARIMA (p,d,q)模型预测出周末、节假日这类特殊变化过程的日负荷值,算出残差序列

,（式5.21）

其中

.（式5.22）

2.若k≥k0时，符号一致，k=k0,k0+1,…，n，则对残差序列

（式5.23）

利用GM(1，1)中的方法求出预测值

，（式5.24）

其中

.（式5.25）

* 的预测修正值：
* ,i≥k0，（式5.26）
* 其中“±”的取定应与残差(k≥k0)的符号一致。即为最终预测结果值。

3.应用效果

将上述算法用计算机编程实现，经运行检验，精确度满足用户要求。

表2为某省电网2000年国庆节连续3日的负荷预测日报表，为了压缩篇幅，10月2日及10月3日只列出了日预测负荷准确率等最后三项，预测数据删去。

表5.2 负荷预测日报表

Table 5.2 Load forecasting statements Japan

|  |  |  |  |
| --- | --- | --- | --- |
| 时 间 | 预测负荷 | 实际负荷 | 相对误差（%） |
| 01:00 | 1934 | 2031 | 5.01 |
| 02:00 | 1960 | 1905 | 2.08 |
| 03:00 | 1868 | 1866 | 0.10 |
| 04:00 | 1863 | 1848 | 0.80 |
| 05:00 | 1849 | 1872 | 1.24 |
| 06:00 | 1934 | 2016 | 4.23 |
| 07:00 | 1922 | 2155 | 6.17 |
| 08:00 | 2172 | 2162 | 0.46 |
| 09:00 | 2321 | 2293 | 1.20 |
| 10:00 | 2373 | 2361 | 0.50 |
| 11:00 | 2462 | 2405 | 2.31 |
| 12:00 | 2264 | 2250 | 0.61 |
| 13:00 | 2251 | 2166 | 3.77 |
| 14:00 | 2274 | 2179 | 4.17 |
| 15:00 | 2256 | 2252 | 0.17 |
| 16:00 | 2335 | 2309 | 1.11 |
| 17:00 | 2218 | 2492 | 12.30 |
| 18:00 | 2498 | 2490 | 0.32 |
| 19:00 | 2799 | 2772 | 0.96 |
| 20:00 | 2816 | 2863 | 1.66 |
| 21:00 | 2688 | 2757 | 2.56 |
| 22:00 | 2495 | 2608 | 4.52 |
| 23:00 | 2213 | 2269 | 2.53 |
| 24:00 | 2018 | 2063 | 2.22 |
| 日预计负荷准确率（%）：96.5 |  | |  |
| 日最高负荷预计： 2816 | 实际：2863 | | 准确率（%）： 98.3 |
| 日最低负荷预计： 1849 | 实际：1848 | | 准确率（%）： 99.9 |
| 单位：MW |  | | 日期：2000-10-2 |
| 日预测负荷准确率（%）：96.0 |  | |  |
| 日最高负荷预测： 2977 | 实际：2871 | | 准确率（%）： 96.4 |
| 日最低负荷预测： 1803 | 实际：1844 | | 准确率（%）： 97.7 |
| 单位：MW |  | | 日期：2000-10-3 |
| 日预测负荷准确率（%）：95.2 |  | |  |
| 日最高负荷预测： 3151 | 实际：3008 | | 准确率（%）： 95.4 |
| 日最低负荷预测： 1885 | 实际：1855 | | 准确率（%）： 98.4 |

5.3.3 对气候温度剧变日负荷预测值的处理

采用前述方法，对预测周末、节假日这类变化过程的日负荷值具有较好的准确率。而对预测气候温度剧变这类变化过程的日负荷值准确率虽有一定的提高，但是仍不能完全满足用户提出的“预测准确率达到95%以上的要求”。在对试算结果进行统计分析中发现：当预测周末、节假日的日负荷值时，如果正值气候温度剧变，预测准确率也会下降。对此，根据气候温度的剧变程度给出不同的调整权重，如气候分“晴、阴、雨”；温度以28℃为分界，低于28℃每相差4℃为一档，高于28℃每相差2℃为一档。为了便于分析，我们定义了“气候温度同时差与“相对气候温度同时差”：

“气候温度同时差”—上一年与预测点同月同日类气候温度条件下的同时刻预测值与实际值的差值。

“相对气候温度同时差”—上一年与预测点前后各348个取样点(约15天)中类气候温度条件下的首点的预测值与实际值的差值。

“类气候温度条件”指的是：气候温度的变化在划定的同一档内。

“气候温度剧变”—预测日前是晴雨交替或日平均温度跳档。

对气候温度剧变条件下的预测值按以下原则处理：先判断是否存在气候温度同时差，如存在，将气候温度同时差值除以实际值得到差比，然后将预测值加上差比×预测值，得到最后的预测值；如不存在，转求相对气温同时差及差比，经同样处理后得到最后的预测值。预测结果大部分达到用户的要求。

## 5.4 类图

类图一般在详细设计过程中出现，主要用来描述系统中各个模块中类之间的关系，包括类或者类与接口的继承关系，类之间的依赖、聚合等关系。它还描述每一个类的详细信息，包括变量，和方法。通过类图，就能把系统中的各个类，即对象描述清楚，然后就是按照这个详细的设计编码了。

### 5.4.1 实体类图

如图5.1所示是该系统的实体类图。

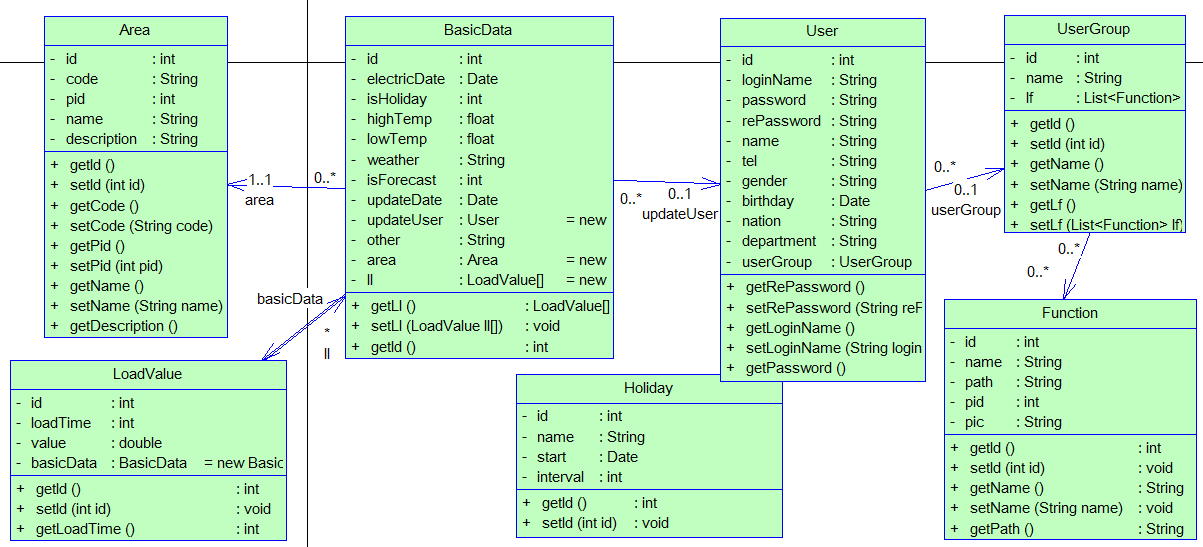


图5.1 短期电力负荷预测系统实体类图

Fig.5.1 The entity class diagram of short term electric load forecasting system

实体类存放在包model当中，这些类充当数据库中数据的载体，对应于数据库的表，在编码阶段的每个类当中都需要用到这些实体，它们是整个系统的基础。

### 5.4.2 控制类图

如图5.2所示是该系统的控制类图。

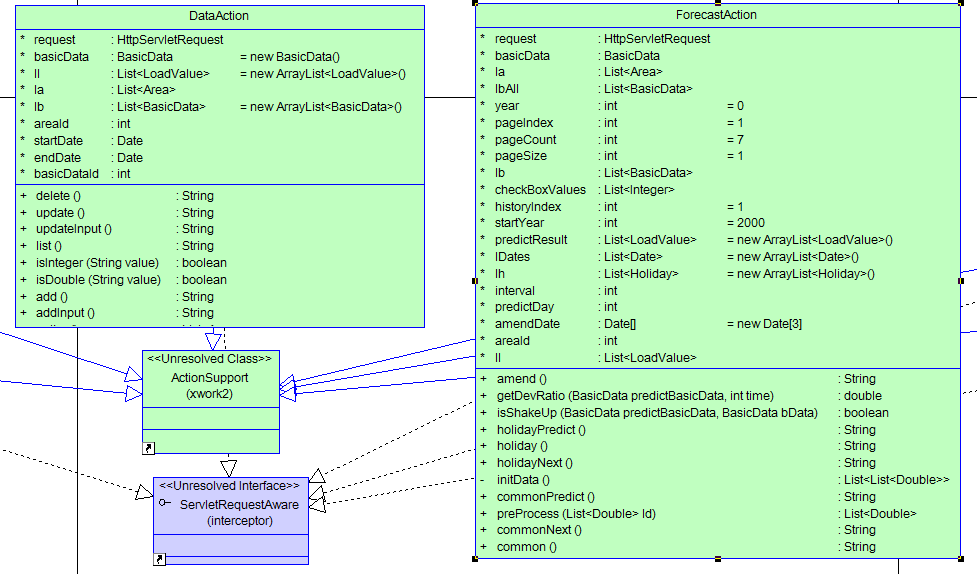


图5.2 短期电力负荷预测系统控制类图

Fig.5.2 Control class diagram of short term power load forecasting system

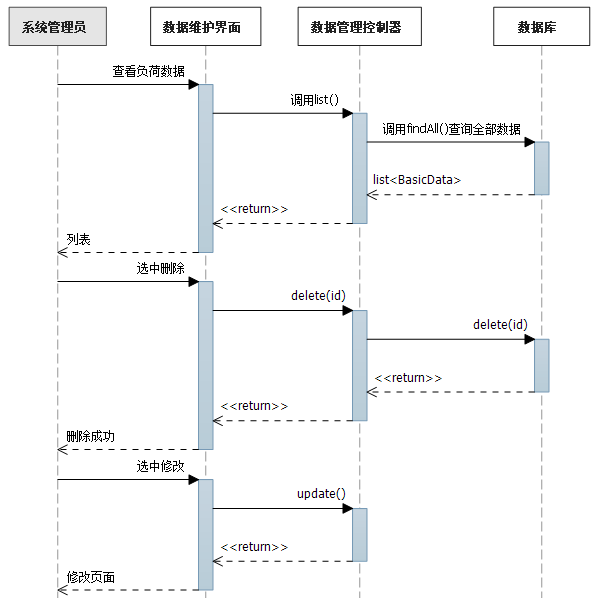
在struts2中，控制类相当于action类，如图5.2所示，每个action继承了ActionSupport类并且实现了ServletRequestAware。限于篇幅这里只列出了俩个核心action：DataAction与ForecastAction。DataAction主要实现对历史数据的操作，ForecastAction主要实现负荷预测的功能。

## 5.5 时序图

时序图（Sequence Diagram），亦称为[序列图](http://baike.baidu.com/view/2226296.htm)或循序图或顺序图，是一种UML交互图。它通过描述[对象](http://baike.baidu.com/view/2387.htm)之间发送[消息](http://baike.baidu.com/view/89742.htm)的时间顺序显示多个对象之间的动态协作。它可以表示[用例](http://baike.baidu.com/view/706238.htm)的行为顺序，当执行一个用例行为时，时序图中的每条[消息](http://baike.baidu.com/view/89742.htm)对应了一个类操作或[状态机](http://baike.baidu.com/view/1906565.htm)中引起转换的触发事件。本文以系统管理员所涉及到的时序图为例，详细描述了系统的实现过程。

### 5.5.1 数据维护时序图

如图5.3所示是该系统的数据维护时序图。数据维护指的是对历史负荷数据的维护，管理员进入数据维护的界面后，系统会调用控制器当中的方法查询数据库中的全部的数据信息并返回到页面上，管理员也可以按照条件进行查询相关数据。点击删除按钮，页面会通过get方式提交到后台，把相关ID传给控制器，控制器根据基本负荷数据的ID在数据库中查询并且删除相关记录。点击修改按钮，控制器返回修改页面，用户修改相应的信息然后再通过控制器提交给数据库。



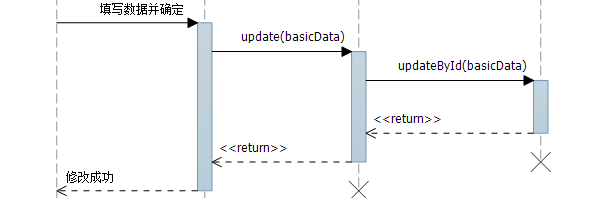


图5.3 负荷数据维护时序图

Fig.5.3 Load data maintenance sequence diagram

### 5.5.2 数据导入时序图

如图5.4所示是该系统的数据导入时序图。管理员填写表单数据，或者选取excel文件后，系统会自动在数据管理控制器里转换为BasicData类型的数据，在添加到数据库之前，控制器先判断该数据是否是重复数据，然后再添加到数据库中。

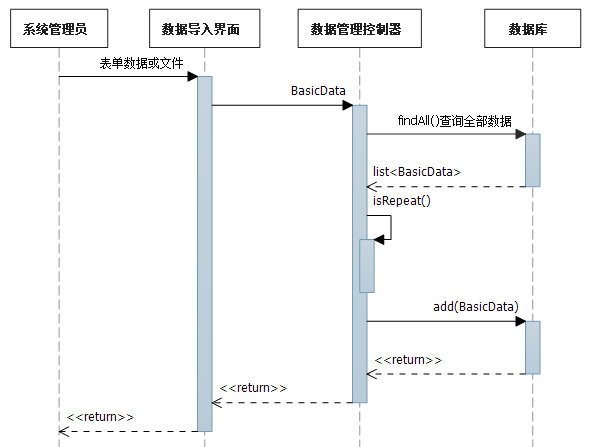


图5.4 负荷数据导入时序图

Fig.5.4 Load data import sequence diagram

### 5.5.3 负荷曲线时序图

如图5.5所示是该系统的负荷曲线时序图。管理员首先查询想要查看曲线的数据，将地区与日期作为参数传给负荷分析控制器，控制器调用相应的方法查询数据库然后返回给管理员。管理员进入曲线查看界面，jsp页面自动调用jQuery的方法，将预先的查询到的数据转换为折线图显示出来。

### 5.5.4 负荷特性时序图

如图5.6所示是该系统的负荷特性时序图。系统管理员首先要按条件查询数据，然后负荷分析控制器调用方法查询数据库后返回一个list集合，控制器再对此集合进行计算处理就产生了负荷特性的数据，最后返回给页面供用户查看。用户选择导出后调用控制器的download方法，控制器调用浏览器的接口产生下载页面。

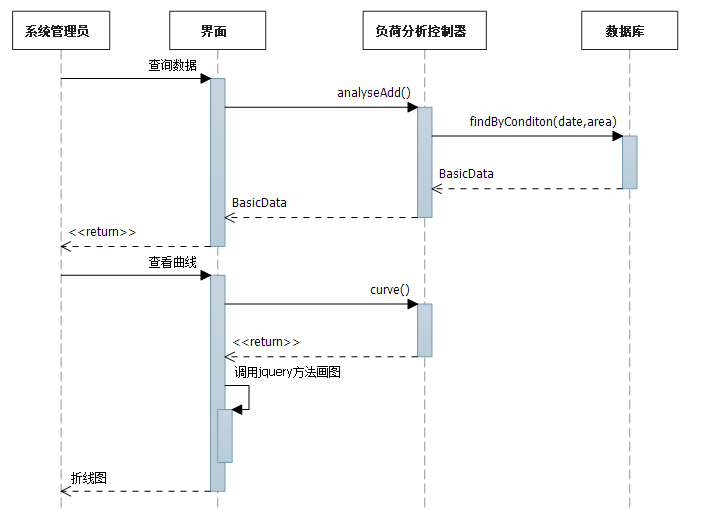


图5.5 负荷曲线时序图

Fig.5.5 load curve sequence diagram

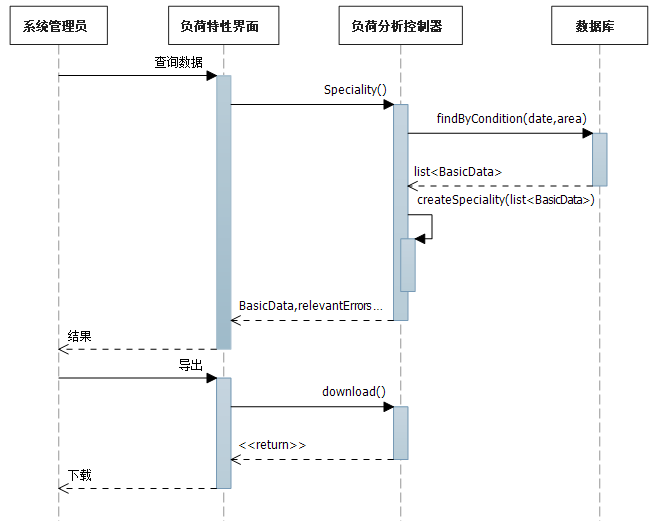


图5.6 负荷特性时序图

Fig.5.6 load speciality sequence diagram

### 5.5.5 正常日预测时序图

如图5.7所示是该系统的正常日预测时序图。系统管理员进入正常日预测界面，填写基本负荷数据后提交，控制器选择下一步的预测界面返回。管理员继续填写信息，向控制器发出查询信号，控制器调用方法查询数据库的历史负荷数据，并且将查询到的数据返回给页面。然后用户选择预测所需要的历史数据，控制器获得用户选择的数据后构造灰色模型，调用灰色模型的predict方法进行预测，然后将灰色模型类返回的数据构造为BasicData对象，并且将其添加到数据库当中。

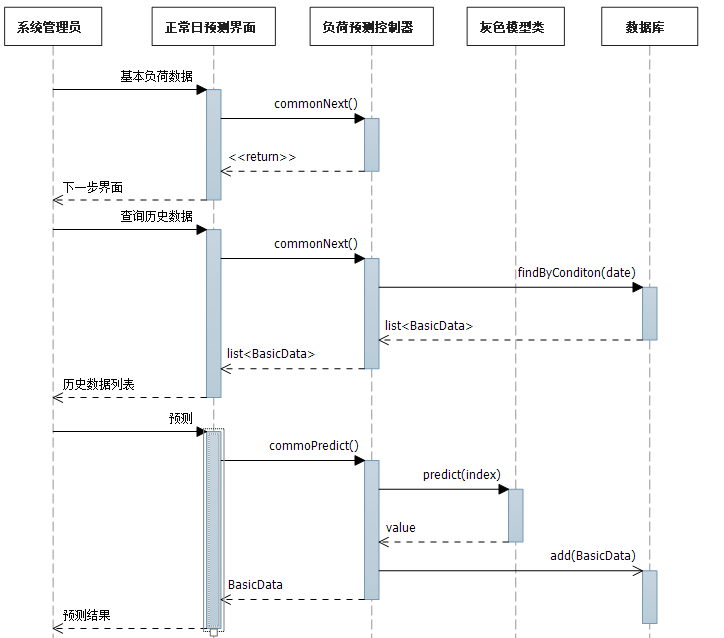


图5.7 正常日预测时序图

Fig.5.7 common day forecast sequence diagram

### 5.5.6 节假日预测时序图

如图5.8所示是该系统的节假日预测时序图。系统管理员进入节假日预测界面，和普通预测界面提交的负荷数据有所不同，节假日预测只能选择节假日作为预测日期，填写基本负荷数据后提交，控制器选择下一步的预测界面返回。管理员继续填写信息，向控制器发出查询信号，控制器调用方法查询数据库的历史负荷数据，并且将查询到的数据返回给页面。然后用户选择预测所需要的历史数据，控制器获得用户选择的数据后构造改进的灰色模型，调用改进灰色模型的predict方法进行预测，然后将灰色模型类返回的数据构造BasicData对象，并且将其添加到数据库当中。

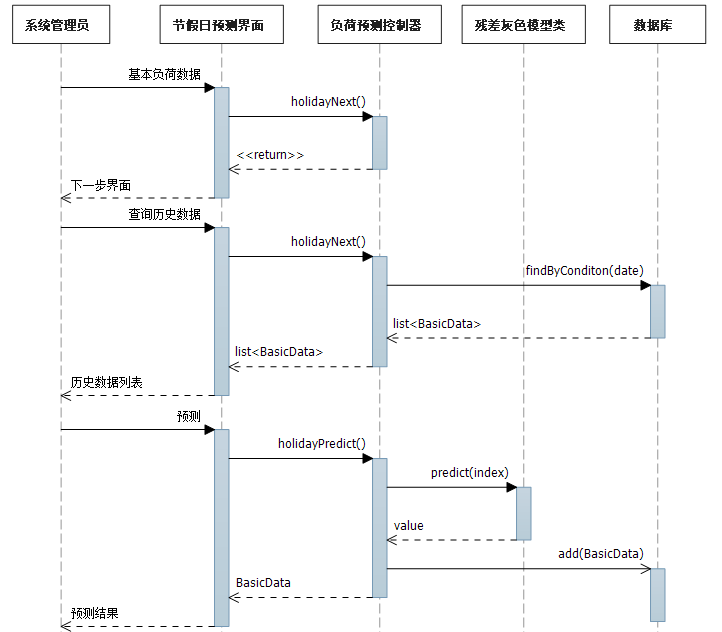


图5.8 节假日预测时序图

Fig.5.8 holiday forecast sequence diagram

### 5.5.7 修正上报时序图

如图5.9所示是该系统的修正上报时序图。系统管理员输入要查询的预测数据以及参照日的信息，控制器类获得参数后，调用方法去数据库进行查询，返回List<BasicData>类型的数据，并且由控制器传到页面。管理员对已经显示出来的数据进行修改，然后由相应的控制器调用修改方法修改数据库中的数据。

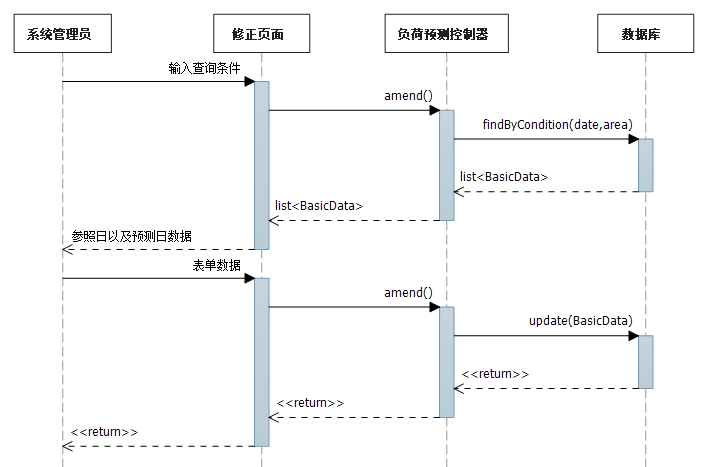


图5.9 修正上报时序图

Fig.5.9 Fixed report sequence diagram

### 5.5.8 账户维护时序图

如图5.10.1与图5.10.2所示是该系统的账户维护时序图。

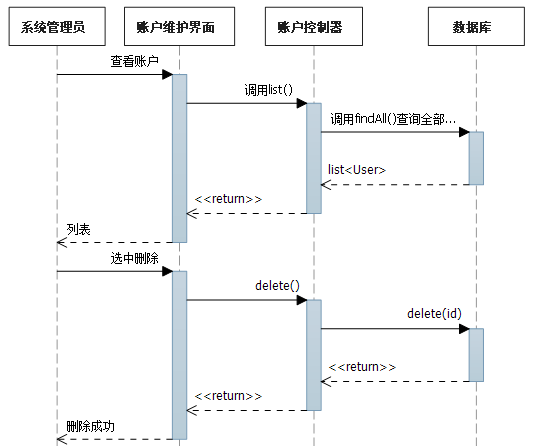


图5.10.1 账户维护时序图

Fig.5.10.1 User maintenance sequence diagram

管理员进入账户维护的界面后，系统会调用控制器当中的方法查询数据库中的全部的数据信息并返回到页面上，管理员也可以按照条件进行查询相关数据。点击删除按钮，页面会通过get方式提交到后台，把相关ID传给控制器，控制器根据账户的ID在数据库中查询并且删除相关记录。选中修改按钮，控制器返回修改页面，用户修改相应的信息然后再通过控制器提交给数据库。管理员进入添加页面提交表单，控制器先通过调用方法判断是否重复，然后再添加账户数据。

关于地区维护，节假日维护等时序图与账户维护的时序图类似，涉及到的都是页面向控制器发出增删查改等命令，然后控制器再进一步进行操作，本文不再赘述。

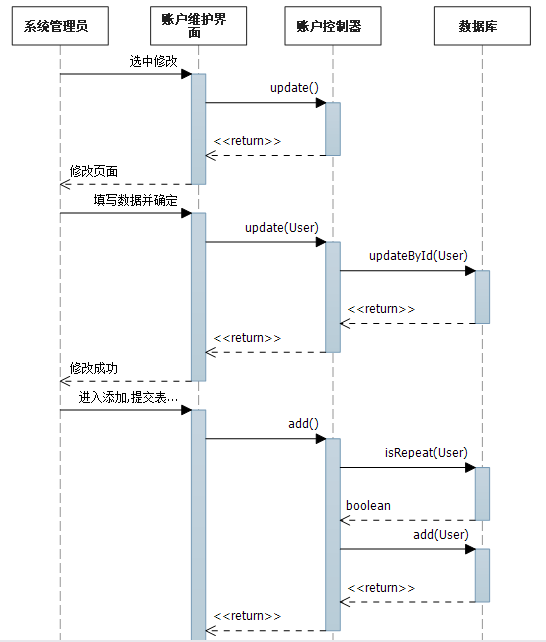


图5.10.2 账户维护时序图

Fig.5.10.2 User maintenance sequence diagram

# 第六章 系统实现与代码编写

## 6.1 系统实现

系统实现与代码编写阶段是将前边所有的逻辑设计转换成用户可操作的系统，这一阶段主要的任务是代码编写，从而产生物理文件。本章将挑选典型的功能模块以及相关代码进行说明。

### 6.1.1 用户登录

用户登录界面如图6.1与图6.2所示。



图6.1 用户登录界面

Fig.6.1 User login interface

验证码是通过<img>标签来实现的。首先在登录页面加入<img>标签，然后将该标签的href属性设置成一个动态的jsp页面，其中的jsp页面包含了图形描绘的代码，将画好的图形以流的形式输出就在登录页面呈现出了验证码。在登录成功界面当中，左边的菜单栏是根据登录账户的不同角色从数据库动态读入。

实现详情如表6.1所示

表6.1 用户登录实现

Table6.1 User login implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **事件** | **Action** | **入口** | **出口** | **使用模型** |
| 登录 | UserAction.java | login.jsp | firstPage.jsp | User.java  UserGroup.java  Function.java |



图6.2 登录成功界面

Fig.6.2 Login successfully interface

### 6.1.2 数据管理

数据管理界面如图6.3与图6.4所示。

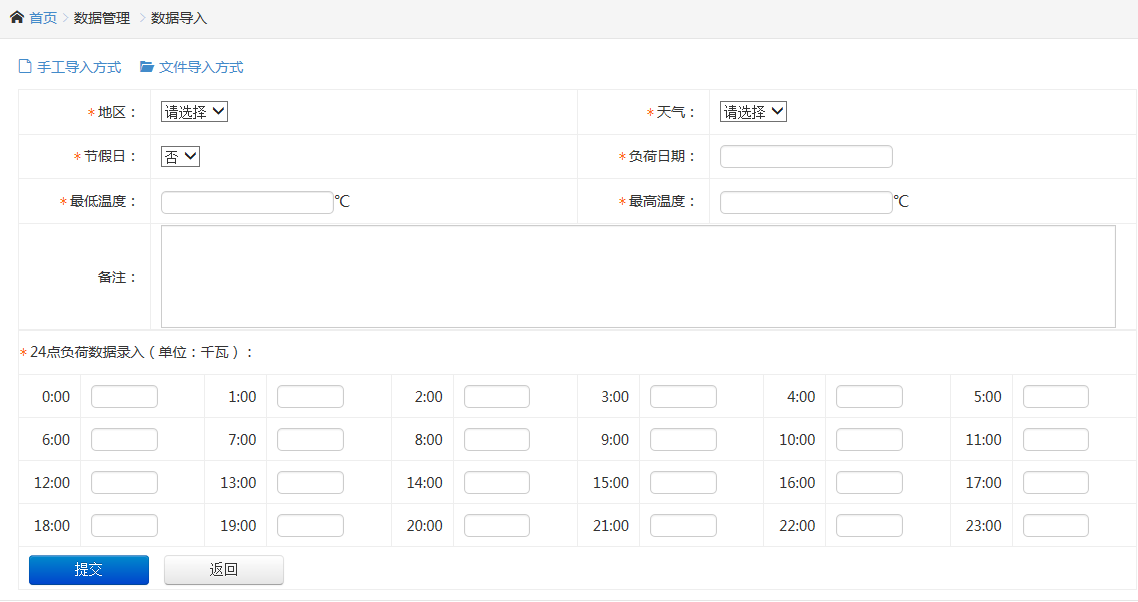


图6.3 导入数据界面

Fig.6.3 Import data interface



图6.4.1 数据维护界面

Fig.6.4.1 Data maintenance interface



图6.4.2 数据维护界面

Fig.6.4.2 Data maintenance interface

导入数据实际上是一个添加负荷数据的过程，手工导入是指有操作人员手动添加负荷基本数据以及24点负荷数据；文件导入是指操作人员选择已有的并且按照规定格式录入数据的Excel文件，将数据导入到数据库当中。在数据维护界面当中，用户可以查找指定日期之间的负荷数据，然后每一行有删除和修改的操作。

实现详情如表6.2所示

表6.2 数据管理实现

Table6.2 Data management implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **事件** | **Action** | **入口** | **出口** | **使用模型** |
| 数据导入 | DataAction.java | data\_addInput.jsp | data\_success.jsp | BasicData.java  LoadValue.java  Holiday.java  Area.java |
| 数据维护 | DataAction.java | data\_list.jsp | data\_updateInput.jsp  data\_success2.jsp | BasicData.java  LoadValue.java  Area.java |

### 6.1.3 负荷分析

负荷分析界面如图6.5与图6.6所示。



图6.5.1 负荷曲线界面

Fig.6.5.1 Load curve interface

在负荷曲线界面第一个界面中，用户通过查询来添加要查看相关曲线的负荷数据，每次查询得到的负荷数据将被存放在session对象当中保持下来，在下次添加的过程中将新得到的负荷数据再次放到session对象里，然后读取session将查询过的负荷数据全部显示出来。用户点击查看曲线，程序再次读取session对象，将读取出来的List<BasicData>数据转换为json，通过jQuery调用<canvas>标签相关的api，实现了绘图的功能。在负荷特性界面中，系统首先查询到用户所需要的实际负荷数据与当天的预测负荷数据，按照负荷特性的定义进行数据处理，得到最后的结果。

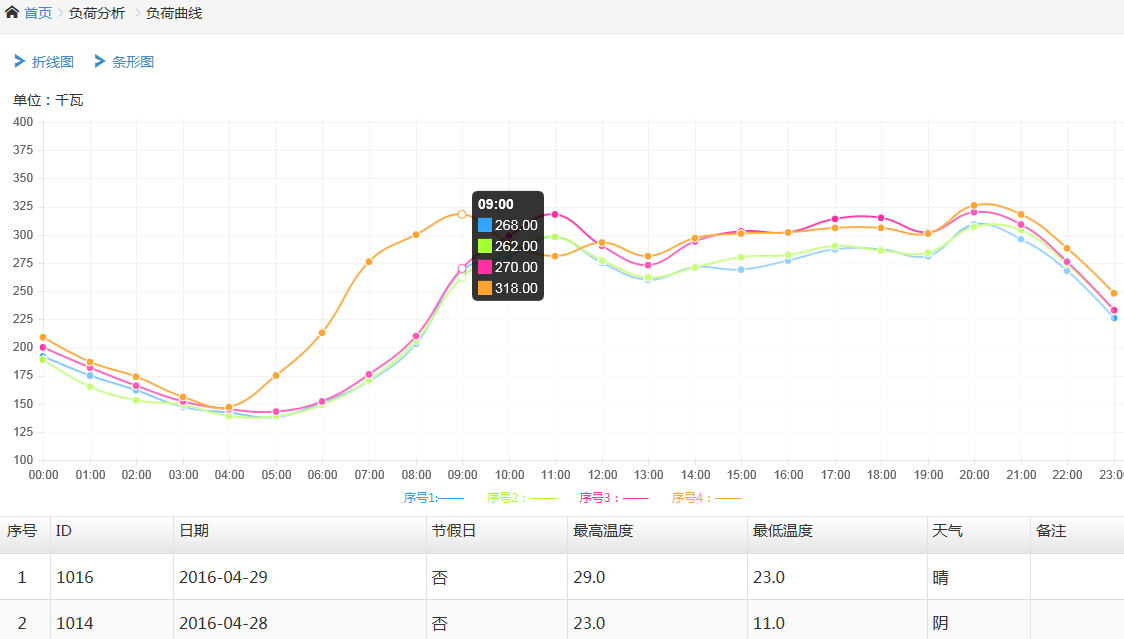


图6.5.2 负荷曲线界面

Fig.6.5.2 Load curve interface



图6.6 负荷特性界面

Fig.6.6 Load speciality interface

实现详情如表6.3所示

表6.3 负荷分析实现

Table6.3 Load analysis implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **事件** | **Action** | **入口** | **出口** | **使用模型** |
| 负荷曲线查看 | AnalyseAction.java | analyse\_add.jsp | analyse\_curve.jsp | BasicData.java  LoadValue.java  Holiday.java  Area.java |
| 数据特性查看 | AnalyseAction.java | analyse\_speciality.jsp | analyse\_speciality.jsp | BasicData.java  LoadValue.java  Holiday.java  Area.java |

### 6.1.4 负荷预测

负荷预测界面如图6.7、图6.8与图6.9所示。



图6.7.1 正常日预测界面

Fig.6.7.1 Load speciality interface

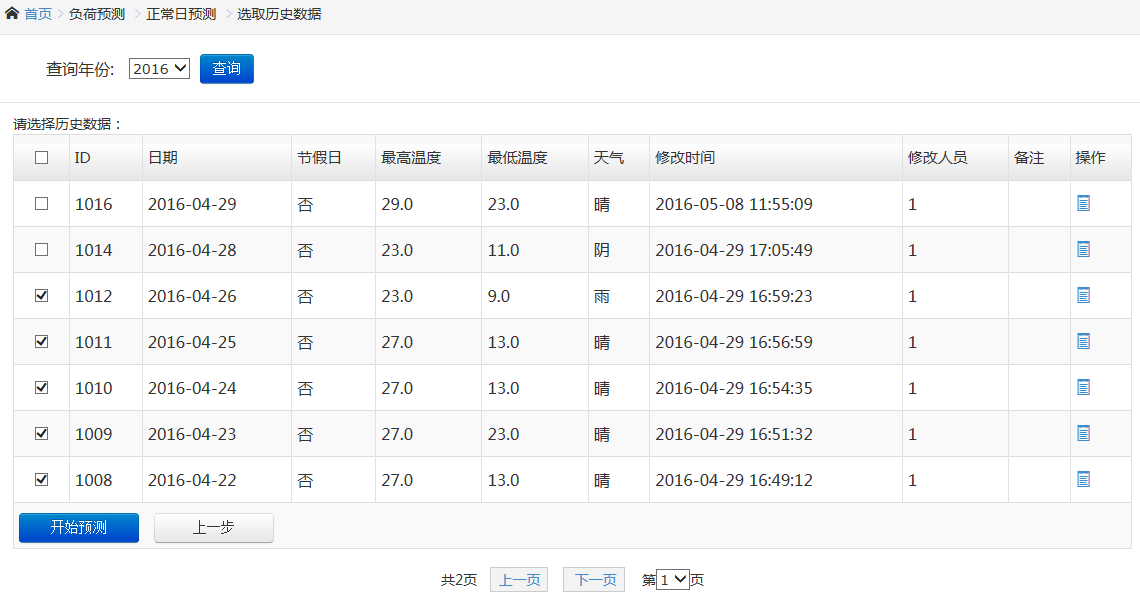


图6.7.2 正常日预测界面

Fig.6.7.2 common day forecast interface



图6.7.3 正常日预测界面

Fig.6.7.3 common day forecast interface



图6.8 节假日预测界面

Fig.6.8 holiday forecast interface

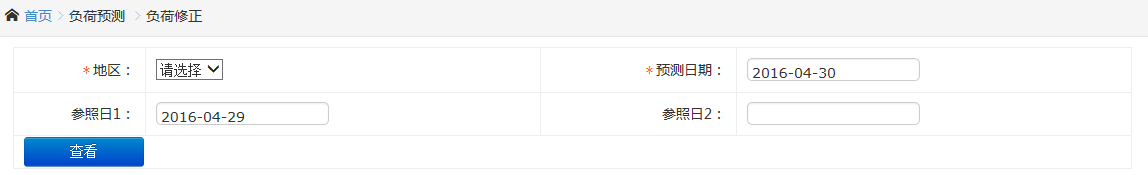


图6.9.1 修正上报界面

Fig.6.9.1 Fixed report holiday forecast interface

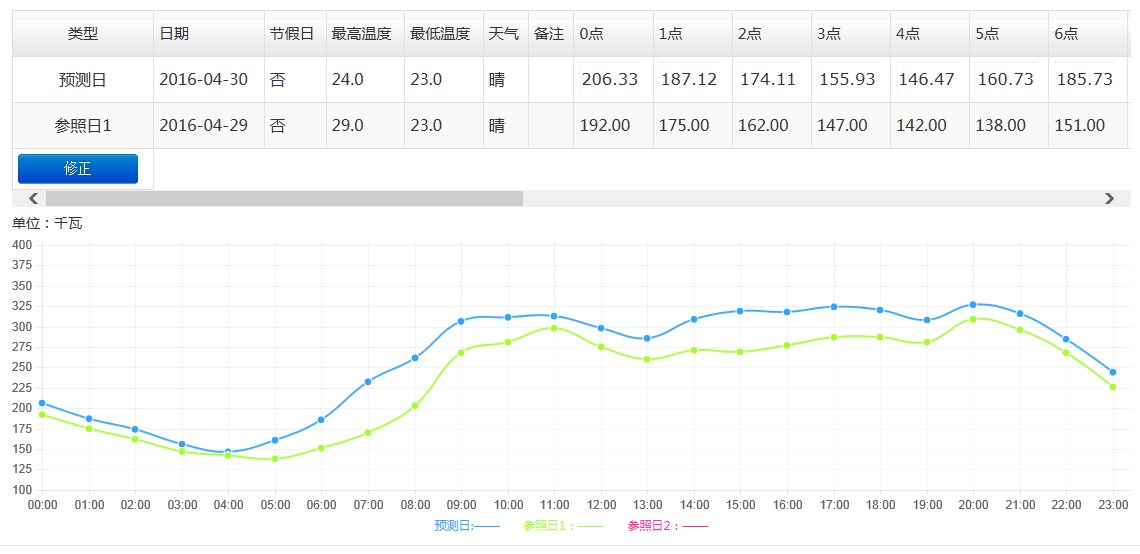


图6.9.2 修正上报界面

Fig.6.9.2 Fixed report holiday forecast interface

进行普通日预测的时候首先填写预测的基本数据信息，点击进入下一步页面，此时由session保存刚刚用户提交的数据。在下一步页面当中，系统查询历史负荷数据，用户可以自由选择进行预测所需要的基准数据。程序先将数据全部从数据库读出来放在List里，使用PageBean类实现分页功能。用户点击预测以后，系统首先将选择的数据通过滑动平均法进行数据预处理，然后构造GM(1,1)模型进行24点预测。节假日预测的界面和普通日预测的界面类似，但是选择的日期必须是节假日，且预测的方法是GM(1,1)改进模型。在修正上报当中，用户可以选择参考日来对预测数据进行手动修正。

实现详情如表6.4所示

表6.4 负荷预测实现

Table6.4 Load forecast implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **事件** | **Action** | **入口** | **出口** | **使用模型** |
| 正常日预测 | ForecastAction.java | forecast\_common.jsp | forecast\_commonNext.jsp  forecast\_commonPredict.jsp | BasicData.java  LoadValue.java  Area.java |
| 节假日预测 | ForecastAction.java | forecast\_holiday.jsp | forecast\_holidayNext.jsp  forecast\_holidayPredict.jsp | BasicData.java  LoadValue.java  Holiday.java  Area.java |
| 修正上报 | ForecastAction .java | forecast\_amend.jsp | analyse\_amend.jsp | BasicData.java  LoadValue.java  Holiday.java  Area.java |

### 6.1.5 系统设置

系统设置界面如图6.10与图6.11所示。

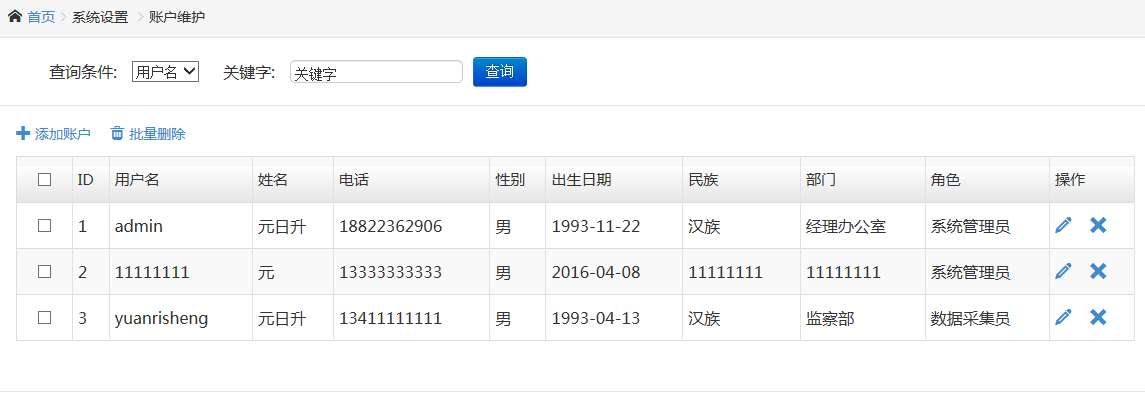


图6.10 账户维护界面

Fig.6.10 User maintenance interface

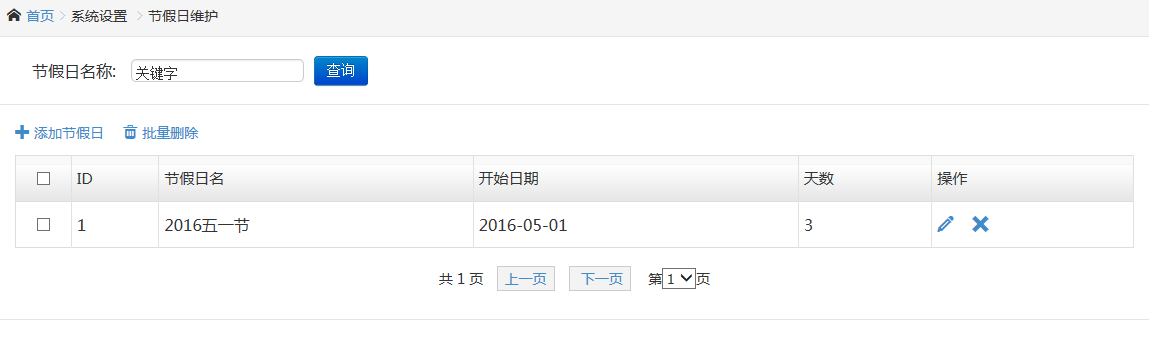


图6.11 节假日维护界面

Fig.6.11 Holiday maintenance interface

系统设置主要是对系统的非核心功能的操作，其中包括账户，节假日，地区维护，每个模块都可以对相应的数据进行增删查改。

实现详情如表6.5所示

表6.5 系统设置实现

Table6.5 System setting implementation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **事件** | **Action** | **入口** | **出口** | **使用模型** |
| 账户维护 | UserAction.java | user\_list.jsp | user\_addInput.jsp  user\_updateInput.jsp  user\_success.jsp | User.java |
| 节假日维护 | HolidayAction.java | holiday\_list.jsp | holiday\_addInput.jsp  holiday\_updateInput.jsp  holiday\_success.jsp | Holiday.java |
| 地区维护 | AreaAction .java | area\_list.jsp | area\_addInput.jsp  area\_updateInput.jsp  area\_success.jsp | Area.java |

# 第七章 测试

软件测试是软件项目的质量控制中最常用的手段之一，也是在软件开发过程中比较容易被忽略和执行得不完善和不彻底的工作。

## 7.1 测试的目的

软件测试的目的是尽可能发现并改正被测试软件中的错误，提高软件的可靠性。测试涉及众多的人员，不同的人从不同的视角来对待测试，主要包括以下几个方面：在实现一个程序单元时，程序员可能想要测试在正常环境下单元能否工作。如果单元工作令他满意，程序员将获得很大信心。一旦程序员对单元在一定程度上的有效工作满意，他将会进行更多的测试，发现单元或者整个系统的故障。其次，大多数复杂的软件系统都包含故障，它们可能会引起系统失败。因此，实施测试的一个更高的目标就是降低失败的风险。最后，软件的测试也需要成本，实施测试的最高级别的目标就是使用数量更少的测试用例来生产低风险的软件。

## 7.2 测试过程

此次测试的过程中包含了单元测试，集成测试，确认测试和系统测试。使用的最主要的黑盒测试法。

本系统的测试，通过手动操作该系统，主要查看的是是否存在异常或操作的结果是否按照设计初衷来完成的，主要分数据管理模块，负荷分析模块以及负荷预测模块，系统管理来测试。

### 7.2.1 数据管理测试

数据管理测试的过程如表7.1与表7.2所示。

表7.1 数据导入测试用例

Table7.1 Data import test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 数据导入 | | | **编制人** | 元日升 | | |
| **用例编号** | 数据管理\_1 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 对历史负荷数据进行导入 | | | | | | |
| **测试目的** | 验证是否输入合法的信息，确保数据完整性，以及确保数据写入数据库 | | | | | | |
| **预置条件** | 用户收集到相关负荷数据 | | | **特殊规程说明** | 负荷单位是千瓦，温度单位是摄氏度 | | |
| **参考信息** | 需求分析中关于“数据管理”的说明 | | | | | | |
| **测试数据** | 地区=“河西区” | | 天气=“晴” | | | 节假日=“否” | |
| 负荷日期=“2016-04-29” | | 备注=“” | | | 最低温度=23 | |
| 最高温度=25 | | 0点负荷=192 | | | 1点负荷=175 | |
| 2点负荷=162 | | 3点负荷=147 | | | 4点负荷=142 | |
| 5点负荷=138 | | 6点负荷=151 | | | 7点负荷=170 | |
| 8点负荷=203 | | 9点负荷=268 | | | 10点负荷=281 | |
| 11点负荷=298 | | 12点负荷=275 | | | 13点负荷=260 | |
| 14点负荷=271 | | 15点负荷=269 | | | 16点负荷=277 | |
| 17点负荷=287 | | 18点负荷=287 | | | 19点负荷=281 | |
| 20点负荷=309 | | 21点负荷=296 | | | 22点负荷=268 | |
| 23点负荷=226 | |  | | |  | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 直接点击提交。 | 都为空 | | 显示警告信息“地区不能为空！”、“天气不能为空！”、“负荷日期不能空！”、“负荷数据不能为空！” | （符合） | | P |
| 2 | 在温度和电负荷表单中填写汉字 | 最高温度=“你好”，1点负荷=“你好” | | 显示警告信息“请填写合法数字！” | （符合） | | P |
| 3 | 填写表单，点击提交 | 测试数据 | | 添加成功 | （符合） | | P |

表7.2 数据维护测试用例

Table7.2 Data maintenance test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 数据维护 | | | **编制人** | 元日升 | | |
| **用例编号** | 数据管理\_2 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 对历史数据进行查询，删除以及修改 | | | | | | |
| **测试目的** | 确保读写历史数据正确 | | | | | | |
| **预置条件** | 数据库表BasicData，LoadValue有数据 | | | **特殊规程说明** | 负荷单位是千瓦，温度单位是摄氏度 | | |
| **参考信息** | 需求分析中关于“数据管理”的说明 | | | | | | |
| **测试数据** | 地区=“河西区” | | 起始日期=“2016-04-29” | | | 终止日期=“2016-04-28” | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 直接点击查询。 | 都为空 | | 全部负荷数据 | （符合） | | P |
| 2 | 填写地区，起始日期与终止日期点击查询 | 测试数据 | | 河西区2016/04/29与2016/04/28之间的数据 | （符合） | | P |

小结：数据管理基本功能完善，实际结果与期望结果符合。

### 7.2.2 负荷分析测试

负荷分析测试的过程如表7.3与表7.4所示。

表7.3 负荷曲线测试用例

Table7.3 Load curve test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 负荷曲线 | | | **编制人** | 元日升 | | |
| **用例编号** | 负荷分析\_1 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 将负荷数据以曲线的形式展示出来 | | | | | | |
| **测试目的** | 确保曲线显示正常 | | | | | | |
| **预置条件** | 数据库表BasicData，LoadValue有数据 | | | **特殊规程说明** | 负荷单位是千瓦，温度单位是摄氏度 | | |
| **参考信息** | 需求分析中关于“负荷分析”的说明 | | | | | | |
| **测试数据** | 地区=“河西区” | | 日期=“2016-04-29” | | | 负荷性质=“历史数据” | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 填写地区，负荷性质，日期 | 测试数据 | | 在当前页面显示出要查询的负荷数据的信息 | 日期=“2016-04-29” | | P |
| 节假日=“否” | |
| 最高温度=29.0 | |
| 最低温度=23.0 | |
| 天气=“晴” | |
| 备注=“” | |
| 2 | 点击查看曲线 | 操作步骤1得到的数据 | | 打开另一个页面，该页面有一条负荷曲线 | （符合） | | P |

表7.4 负荷特性测试用例

Table7.4 Load speciality test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 负荷特性 | | | **编制人** | 元日升 | | |
| **用例编号** | 负荷分析\_2 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 将负荷数据的日负荷特性显示出来 | | | | | | |
| **测试目的** | 确保正常显示负荷特性 | | | | | | |
| **预置条件** | 数据库表BasicData，LoadValue有数据 | | | **特殊规程说明** | 负荷单位是千瓦，用电量单位千瓦时 | | |
| **参考信息** | 需求分析中关于“负荷分析”的说明 | | | | | | |
| **测试数据** | 地区=“河西区” | | 日期=“2016-04-29” | | |  | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 填写地区，日期，点击查询 | 测试数据 | | 显示出基本负荷信息，当天的24点预测负荷与实际负荷以及相对误差，显示出了日最小负荷，日最大负荷，日峰谷差，日峰谷差率等日负荷特性 | 24电负荷与数据库一致 | | P |
| 日最小负荷=309.00 | |
| 日最大负荷=138.00 | |
| 日峰谷差=171.00 | |
| 日峰谷差率=55.34% | |
| 最小负荷率=44.66% | |
| 日电量=5633.00 | |
| 平均负荷=234.71 | |
| 日负荷率=75.96% | |
| 平均准确率=96.23% | |
| 日最大误差=15.02% | |
|  | |

小结：根据灰色模型预测的数据与实际数据存在一定的偏差，其中有6组数据的误差大于5%，但平均误差小于5%，满足预测的要求。

### 7.2.3 负荷预测测试

负荷预测测试的过程如表7.4与表7.5所示。

表7.5 预测测试用例

Table7.5 Forecast test case

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | | |
| **功能模块** | 正常日预测 | | | **编制人** | 元日升 | | | |
| **用例编号** | 负荷预测\_1 | | | **编制时间** | 2016-5-11 | | | |
| **功能特性** | 实现短期电力负荷预测 | | | | | | | |
| **测试目的** | 确保正确实现GM(1,1)模型算法 | | | | | | | |
| **预置条件** | 数据库表BasicData，LoadValue有数据 | | | **特殊规程说明** | 负荷单位是千瓦，温度单位摄氏度 | | | |
| **参考信息** | 需求分析中关于“负荷预测”的说明 | | | | | | | |
| **测试数据** | 地区=“河西区” | | 日期=“2016-04-29” | | | 天气=“晴” | | |
|  | 最低温度=22 | | 最高温度=28 | | | 备注=“” | | |
|  | | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | | **测试状态（P/F）** |
| 1 | 填写地区，日期，天气，最低温度，最高温度，备注，点击下一步 | 测试数据 | | 进入下一步的预测页面 | （符合） | | | P |
| 2 | 选择前五条历史负荷数据作为预测的基准数据 | 选中的历史数据以及操作步骤1填写的数据 | | 产生24点预测负荷 | 0点=187.2 | | 1点=163.86 | P |
| 2点=151.54 | | 3点=150.13 |
| 4点=138.05 | | 5点=130.93 |
| 6点=130.38 | | 7点=144.46 |
| 8点=176.11 | | 9点=242.2 |
| 10点=281.0 | | 11点=302.6 |
| 12点=275.54 | | 13点=259.45 |
| 14点=268.5 | | 15点=272.83 |
| 16点=273.93 | | 17点=282.79 |
| 18点=283.38 | | 19点=283.49 |
| 20点=300.42 | | 21点=295.86 |
| 22点=267.21 | | 23点=222.96 |

表7.6 修正上报测试用例

Table7.6 Fixed report test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 修正上报预测 | | | **编制人** | 元日升 | | |
| **用例编号** | 负荷预测\_2 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 对预测负荷进行人工修正 | | | | | | |
| **测试目的** | 是否能正常对数据库数据进行修改 | | | | | | |
| **预置条件** | 数据库表BasicData，LoadValue有数据 | | | **特殊规程说明** | 负荷单位是千瓦，温度单位摄氏度 | | |
| **参考信息** | 需求分析中关于“负荷预测”的说明 | | | | | | |
| **测试数据** | 地区=“河西区” | | 日期=“2016-04-29” | | | 参照日1=“2016-04-28” | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 填写地区，日期，天气，参照日，点击查看 | 测试数据栏中的数据 | | 显示出了该天以及参照日的负荷数据以及曲线图 | （符合） | | P |
| 2 | 修改预测日中的某点的负荷值，点击修正 | 0点=185 | | 页面刷新，预测日中的0点数据变为185 | （负荷） | | P |

小结：系统的核心功能正常实现，通过GM(1,1)算法得到了想要的预测数据。

### 7.2.4 系统管理测试

负荷预测测试的过程如表7.7所示。

表7.7 账户维护测试用例

Table7.7 User maintenance test case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **项目/软件** | 短期电力负荷预测系统 | | | **程序版本** | 1.0.0 | | |
| **功能模块** | 账户维护 | | | **编制人** | 元日升 | | |
| **用例编号** | 系统管理 | | | **编制时间** | 2016-5-11 | | |
| **功能特性** | 对用户进行查询，删除以及修改 | | | | | | |
| **测试目的** | 确保读写历史数据正确 | | | | | | |
| **预置条件** | 数据库表t\_User,t\_Function,UserGroup有数据 | | | **特殊规程说明** | 系统使用用户名登录 | | |
| **参考信息** | 需求分析中关于“系统管理”的说明 | | | | | | |
| **测试数据** | 查询条件=“用户名” | | 关键字=“i” | | |  | |
|  | | | | | | | |
| **操作步骤** | **操作描述** | **数据** | | **期望结果** | **实际结果** | | **测试状态（P/F）** |
| 1 | 输入查询条件以及关键字 | 测试数据 | | 查到用户admin以及yuanrisheng | （符合） | | P |

小结：账户维护的功能正常，可以按照不同条件查询，通过关键字实现了模糊查询的功能。

## 7.3 测试结论

这是测试报告的核心，主要汇总测试各种数据并进行度量，度量包括对测试过程的度量和能力评估、对软件产品的质量度量和产品评估。其中需求覆盖分析以及测试覆盖分析如表7.8以及表7.9所示。

表7.8 需求覆盖分析

Table7.8 Demand coverage analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **功能** | **测试点描述** | **是否测试** | **重要等级** | **是否通过** | **备注** |
| 数据管理 | 对历史负荷数据进行批量添加，查询，删除以及修改。 | 是 | A级 | 通过 | 无 |
| 负荷分析 | 能够查看负荷曲线以及计算负荷特性。 | 是 | A级 | 通过 | 无 |
| 负荷预测 | 实现对未来某一天负荷的预测 | 是 | S级 | 通过 | 该部分测试进行多次测试 |
| 系统设置 | 实现对人员、节假日、地区的增删查改 | 是 | B级 | 通过 | 无 |

说明：这里的重要等级分为S级、A级、B级其中S级是最重要的。由表格可知，需求总数为4部分，测试通过的需求总数为4部分，由需求覆盖率=测试通过需求点/需求总数\*100%可得需求覆盖率为100%。

表7.9 测试覆盖分析

Table7.9 Test coverage analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **功能** | **测试用例数** | **执行数** | **未执行数** | **通过数** | **失败数** |
| 数据管理 | 4 | 2 | 2 | 2 | 0 |
| 负荷分析 | 4 | 2 | 2 | 2 | 0 |
| 负荷预测 | 6 | 2 | 4 | 2 | 0 |
| 系统设置 | 6 | 1 | 5 | 5 | 0 |

说明：由表格可知，数据管理模块测试覆盖率为50%，负荷分析模块测试覆盖率为50%，负荷预测模块测试覆盖率为33%，系统设置模块测试覆盖率为16%，系统总的测试覆盖率为35%，系统的测试通过率为100%。

由测试用例以及最后的分析可以得出，本次测试执行不够充分，但是测试的预期目标都已达成，所有经过测试的功能模块全部通过，没有发现重大的功能缺陷，系统较为安全可靠。但系统可能存在边界值输入产生异常的隐患，需要进行进一步测试。

# 

# 第八章 总结

## 8.1 结论

在这次的毕业设计中，首先我了解到了电力预测的相关知识，明白了其中的全部业务过程，锻炼了自己设计软件的能力。

其次我更加深刻地理解了数学模型对于软件开发的重要性，只有扎实的数学功底，开发某些软件才更加容易。这次我的毕业设计使用的是灰色模型，只有先理解了其中的数学原理，才能结合软件的知识去实现这个系统，充分体现出了软件专业与电力专业，数学专业之间的学科交叉性质。

其次，在数据库设计的过程中，每一个关系模式要最好满足第三范式，这样就能避免插入删除修改异常，减少数据冗余。

在系统实现上，我用的Struts2技术，进一步理解了三层架构之间各层各模块之间的调用关系，对功能进行封装的编程能力也愈发成熟，例如这次我把预测的数学模型以及相关的一些对象进行了很好的封装，实现了代码的可插拔。

在详细设计与代码实现中，我们首先地画出时序图与用例图，对用户和界面所要做的每一次交互有一个清晰的认识，这样在后边的编程过程中才能减少重复修改的次数，才能在做的过程中不盲目，才能有目的地更好地实现所要求的功能。

在设计系统的过程中，要以系统的观点去看待项目，每个模块要留有一定的接口与其他模块衔接，设计的软件功能要考虑全面。

## 8.2 展望

我们经过对系统进行需求分析、设计直到系统的最终实现，大体上可以满足短期电力负荷预测的需要了，但是出于各种原因，系统还可以再做更进一步地优化：

1,本系统预测的算法单一，需要构建高扩展性的系统来实现多种算法对电力负荷的预测。在电力预测方面，可以使用综合预测的技术，因为单一的数学模型有时候还是难以描述事物发展的一些规律，采用综合预测技术可以使得预测结果更加准确，比如该系统可以采用BP神经网络与灰色模型混合算法进行电力预测。

2,本系统缺乏预测结果评估过程，由于该系统预测基准日期的选取是由预测人员等相关技术人员进行选取，在增强预测灵活性的同时也增加了人为干预的因素，一个比较差的技术人员选取的数据得到的结果和一个好的技术人员选取数据得到的结果会有一些差距，所以对最后的预测结果进行评估是本系统需要完善的地方。

3,本系统预测的内容还有欠缺，例如缺乏对国名经济指标以及综合指标相关数据的预测。同时本系统只能对一天的负荷数据进行统计分析，没有实现对一个月的用电数据进行统计。

# 参考文献

[1] 牛东晓，曹树华，卢建昌等编著. 电力负荷预测技术及其应用. 北京：中国电力出版社, 2009.06.

[2] 李鹏.电力负荷短期预测理论及模型研究[D].广州：华南理工大学，2001.

[3] 游仕洪，程浩忠，谢宏.应用模糊线性回归模型预测中长期电力负荷[J].电力自动化设备，2006,26(3)：51-53

[4] 冯燕奎, 赵德奎. JSP实用案例教程[M].北京：清华大学出版社, 2004.70-100

[5] 陈会安. JavaScript基础与实例教程[M].北京：中国电力出版社，2007.

[6] 陈亚辉，[缪勇](http://www.baidu.com/link?url=_IuLj1oNggmxYD69T2Kf87D3eLa-DUrYfRDcQr9Gk4LG2QeBDck71nh41TWhQn9dG13I3L5MI7U6jWN14LmC8XWLKo9MEqofa3G6mPi0Y1G).Struts2+Spring+Hibernate框架技术与项目实战.北京：清华大学出版社，2012.05.

[7] 钟秀玉,李英.基于B/S模式下的现代MIS的信息安全[J].贵阳金筑大学学报,2001,(02)

[8] [张海藩](https://www.amazon.cn/s/ref=dp_byline_sr_book_1?ie=UTF8&field-author=%E5%BC%A0%E6%B5%B7%E8%97%A9&search-alias=books), [牟永敏](https://www.amazon.cn/s/ref=dp_byline_sr_book_2?ie=UTF8&field-author=%E7%89%9F%E6%B0%B8%E6%95%8F&search-alias=books).软件工程导论.北京：清华大学出版社，2013.08

[9] Roger S-Pressman.Software Engineering[M].New Delhi:Tata McGraw-Hill PublishingCompany Ltd,2006.

[10] 颜志军，孙宝文，王天梅.基于UML的业务流程模型分析方法研究.计算机工程与应用，2004

[11] 徐宝文，卢红敏.UML与软件建模.北京：清华大学出版社，2006

[12] Jim Arlow.UML2.0 and the Unified Process[M].机械工业出版社,2006.06

[13] 耿祥义.JSP基础编程[M] .北京：清华大学出版社，2004．55-162

[14] [钱雪忠.数据库原理及应用](https://www.baidu.com/s?wd=%E3%80%8A%E6%95%B0%E6%8D%AE%E5%BA%93%E5%8E%9F%E7%90%86%E5%8F%8A%E5%BA%94%E7%94%A8%E3%80%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuhuhuW-BuWTzPjb1mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmdP1mLnW01)第二版.北京：[北京邮电大学出版社](https://www.baidu.com/s?wd=%E5%8C%97%E4%BA%AC%E9%82%AE%E7%94%B5%E5%A4%A7%E5%AD%A6%E5%87%BA%E7%89%88%E7%A4%BE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuhuhuW-BuWTzPjb1mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmdP1mLnW01),2007.08

[15] 施泉生.短期负荷预报模型库的研究及应用［J］.系统工程理论与实践，1996，16(7)：99-105.

[16] 刘晨晖.电力系统负荷预报理论与方法［M］.哈尔滨：哈尔滨工业大学出版社，1987.

[17] 牛东晓，曹树华，赵磊，等.电力负荷预测技术及其应用［M］.北京：中国电力出版社，1998.

[18] [Ron Patton著. 软件测试.周予滨,姚静译](https://www.baidu.com/s?wd=%E3%80%8A%E6%95%B0%E6%8D%AE%E5%BA%93%E5%8E%9F%E7%90%86%E5%8F%8A%E5%BA%94%E7%94%A8%E3%80%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuhuhuW-BuWTzPjb1mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmdP1mLnW01).[机械工业出版社](https://www.baidu.com/s?wd=%E5%8C%97%E4%BA%AC%E9%82%AE%E7%94%B5%E5%A4%A7%E5%AD%A6%E5%87%BA%E7%89%88%E7%A4%BE&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YvuhuhuW-BuWTzPjb1mWRk0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHmdP1mLnW01),2013.07

# 附 录

1. **灰色模型类**

package com.yuan.predict;

import java.util.ArrayList;

import java.util.List;

import com.yuan.util.Matrix;

public class GrayModel {

// x0为原始参数，x1是根据x0累加得到的参数

protected List<Double> x0 = new ArrayList<Double>();

protected List<Double> x1 = new ArrayList<Double>();

protected double u;

protected double a;

// 1 初始化x0与x1

public GrayModel(List<Double> x0) {

this.x0 = x0;

for (int i = 0; i < x0.size(); i++) {

if (i == 0) {

x1.add(x0.get(i));

} else {

x1.add(x1.get(i - 1) + x0.get(i));

}

}

}

// 2 求B矩阵

protected Matrix getMatrixB() {

double[][] elements = new double[x0.size() - 1][2];

for (int i = 0; i < x0.size() - 1; i++) {

for (int j = 0; j < 2; j++) {

if (j == 0) {

elements[i][j] = -0.5 \* (x1.get(i) + x1.get(i + 1));

} else {

elements[i][j] = 1;

}

}

}

Matrix result = new Matrix(x0.size() - 1, 2, elements);

return result;

}

// 3求向量Y

protected Matrix getMatrixY() {

double[][] elements = new double[x0.size() - 1][1];

for (int i = 0; i < x0.size() - 1; i++) {

elements[i][0] = x0.get(i + 1);

}

Matrix result = new Matrix(x0.size() - 1, 1, elements);

return result;

}

// 4求带有估计值的矩阵U，计算u和a

protected void getUnknownParam() {

Matrix B = getMatrixB();

Matrix Y = getMatrixY();

Matrix BT = Matrix.transpose(B);

Matrix resultMatrix = Matrix.multiply(

Matrix.multiply(Matrix.inverse(Matrix.multiply(BT, B)), BT), Y);

a = resultMatrix.getElements()[0][0];

u = resultMatrix.getElements()[1][0];

// 为了提高精度将矩阵算法恒等变换

// Matrix resultMatrix = Matrix.multiply(

// Matrix.multiply(Matrix.adjoint(Matrix.multiply(BT, B)), BT), Y);

// double det\_val = Matrix.calDet(Matrix.multiply(BT, B).getElements());

// a = resultMatrix.getElements()[0][0] / det\_val;

// u = resultMatrix.getElements()[1][0] / det\_val;

// System.out.println(a + ":" + u);

}

// 5求得预测公式，并且根据t预测t+1

protected double getPredictExp(int t) {

getUnknownParam();

if (t < 0) {

return x0.get(0);

}

double x0tNext = (Math.pow(Math.E, -a) - 1) \* (x0.get(0) - u / a)

\* Math.pow(Math.E, -a \* t);

return x0tNext;

}

// 进行预测

public double predict(int index) {

double result = getPredictExp(index);

return result;

}

}

1. **残差灰色模型类**

package com.yuan.predict;

import java.util.ArrayList;

import java.util.List;

public class ResidualModel extends GrayModel {

private List<Double> xResidual = new ArrayList<Double>();// 残差序列

private int k;// 开始用残差进行修正的位置

public ResidualModel(List<Double> x0) {

super(x0);

// TODO Auto-generated constructor stub

}

// 生成残差序列

private void createRedidual() {

// List<Double> predictX0=new ArrayList<Double>();

for (int i = 0; i < x0.size(); i++) {

// predictX0.add(getPredictExp(i));

double predictx = getPredictExp(i);

xResidual.add(predictx - x0.get(i));

}

}

private void getK() {

boolean flag = true;

for (int i = 0; i < xResidual.size(); i++) {

if (flag) {

if (xResidual.get(i) < 0) {

k = i;

flag = false;

}

} else {

if (xResidual.get(i) >= 0) {

k = i;

flag = true;

}

}

}

}

@Override

public double predict(int index) {

// TODO Auto-generated method stub

createRedidual();

getK();

// 如果需要修正的预测数据太少，按照正常灰色模型预测

if (xResidual.size() - k <= 3)

return super.predict(index);

// 使用残差模型修正预测

List<Double> xChildren = new ArrayList<Double>();

// 提取残差子序列

for (int i = k; i < xResidual.size(); i++) {

xChildren.add(Math.abs(xResidual.get(i)));

}

// 使用灰色模型预测残差

double result = getPredictExp(index);

// 判断是否为需要修正的数据

if (index >= k) {

GrayModel grayModel = new GrayModel(xChildren);

double residual = grayModel.getPredictExp(index - k);

if (xResidual.get(k) > 0) {

result = result - residual;

} else {

result = result + residual;

}

}

return result;

}

}

# 致 谢

时间飞逝，大学四年转瞬即逝，在过去的学习生活中，收获了很多了，而这些成绩的取得是和一直关心帮助我的人分不开的。我需要对在我成长历程中、在人生某个阶段一直给予我帮助、指导、鼓励、支持、信任、爱护的人，表达真正的感谢。

我首先要感谢我的毕业设计指导老师马向国老师给我做出的指导性的意见和建议，在论文撰写过程中及时对我遇到的困难和疑惑给予悉心指点，提出了许多有益的改进性意见，投入了大量的心血和精力。

同时感谢我亲爱的同学们，在学习中我们相互帮助，互相激励和关心。是你们让我在学习和生活中收获到了更多的东西。

最后，我还要感谢我的亲人，谢谢你们在远方对我的关心、支持与鼓励，让远离家乡的我依然感受到温暖。

我始终相信，没有比人更高的山峰，没有比脚印更长的旅程！虽然这次毕业设计已经结束，但学习是永无止境的。我将继续前行，去攀登知识的高峰，踏遍学问的大道，生命不息，奋斗不止！