**北京理工大学现代远程教育学院**

**毕业实践报告**

**银行数据中心运行分析平台**

**设计与实现**

**题目：银行数据中心运行分析平台设计与实现**

**专 业：计算机科学与技术**

**班 级：20140111400**

**姓 名：谢吉星**

**学 号：20140111400009**

**导 师：姜增如**

**日 期：2016年04月**

**学生毕业实践（报告）评语表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 谢吉星 | 学 号 | 20140111400009 | 班 级 | 20140111400 | |
| 第一阶段  （40分） | | | 第二阶段  （60分） | | 合 计  （100分） | |
|  | | |  | | |  |
| 毕业实践（报告）评语  谢吉星同学结合自己得本职工作, 以数据中心运行分析平台系统编程为毕业实践，根据银行数据中心从各个业务系统中抽取的数据，通过ETL处理，编写了实现了前端报表展现的管理功能程序,包括用户管理、机构管理、菜单管理、用户组管理、报表管理、用户登录、用户登出、报表查询等 该程序能够满足大多用户的要求。毕业实践中掌握了系统的业务软件开发整体过程，该毕业实践符合计算机科学专业大专要求，同意毕业。  指导老师（签字）： 年 月 日 | | | | | | |
| 毕业实践（报告）成绩  指导老师组长（签字）： 年 月 日 | | | | | | |

**北京理工大学继续教育暨现代远程教育学院**

**专科学生毕业实践（报告一）**

学生姓名：谢吉星指导教师：姜增如学号：20140111400009 专 业：计算机科学与技术

1. **课题目的和意义**

银行数据中心部门较多，日常运维数据相对分散，各部门数据不能实现集中共享，手工报表较多，日常工作量大，且存在同一指标不同来源数据不一致，全中心汇总数据无来源等情况，随着某行业务的逐步扩展，运维中心的规模及数据量级也逐步攀升，对运维的日常工作也随之提高，原有的低效率、高分散性的工作模式已不能实现新一代系统建设的要求，缺乏有效的数据安全管理机制，数据完备性不足，系统应用推广性差、不具备良好可扩展性, 已不能支持更多的新业务应用发展。

数据中心的报表分散，没有统一的数据集成及统一处理，各类报表都分散在各个应用及运维管理系统中。而且各个部室经常出现报表重复统计的现象。报表中的数据自动生成程度不高，很多是手工录入的。报表的分类标准不明确，没有形成统一的规范标准。各个报表系统各自独立，没有标准化的接口和统一的数据模型，无法复用。报表展现类型单一，需要多维度展现，根据不同用户需求，进行个性化定制。

因此，数据中心需要建设平稳简单的可扩充性和开放性的运行数据分析平台，实现高质量的数据和统一的数据视图，提供报表、灵活查询功能，最终实现深层数据应用。

**二、本课题主要研究内容和预期目标**

**2.1研究内容：**

系统从各个运行监控、管理系统抽取数据，通过ETL处理，装载到运行分析平台的贴源层，然后将数据进行标准化处理，将数据在基础数据层进行存储。在基础数据层之上，建立数据汇总层和应用数据层，用于提供分析应用的数据及报表展现。

**2.2总体功能结构：**

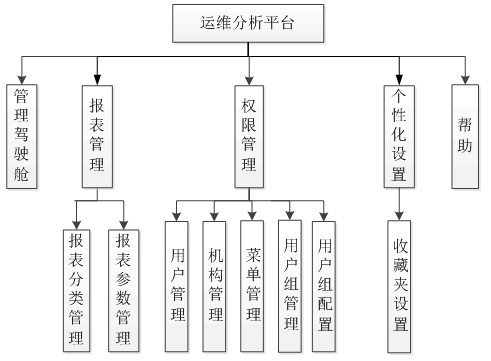


图 2-1 系统功能结构图

**2.3预期目标**：

本毕业设计的内容是设计并且实现银行数据中心运行分析平台，故而系统主要以j2EE作为开发基础,主要使用了spring+ struts+mybatis等技术，用eclipse、PLSQL Developer作为开发工具，以ORACLE作为数据库。基本实现了运行分析平台应有的主要功能模块,包括：数据模型设计、ETL设计、用户管理、机构管理、菜单管理、角色管理、报表管理、用户登录、用户登出、报表查询等功能。该系统界面简单、操作方便，容易维护。

**2.4进度安排：**

2015年12月20日-2015年1月15日 完成开题报告；

2016年1月16日-2016年1月30日 完成系统需求分析；

2016年2月1日-2016年2月14日 完成概要设计、详细设计；

2016年2月15日-2016年2月19日 完成开发环境搭建等开发前准备工作；

2016年2月20日-2016年3月31日 完成程序编码；

2016年4月1日-2016年4月8日 完成平台测试；

2016年4月9日-2016年4月15日 完成论文设计；

2016年4月16日-2016年4月20日 准备答辩；

**三、本课题拟采用的研究方法、步骤**

通过构建统一的业务分析框架，建立全行统一的运行分析环境和基础数据整合平台，来实现对运行中心全方位的管理、分析与决策。

核心数据区是数据仓库数据存储的核心区域，分为缓冲层、基础层和汇总层。

缓冲层与源数据结构一致，主要作用为暂存后为基础层数据整合服务。

基础层参照主流模型设计方法，根据运维中心业务进行主题模型构建，实现基础数据的统一计算、存储和管理。

汇总层参照管理分析业务的需求进行模型构建，实现公共汇总数据的统一计算、存储和管理。

统一报表访问平台将集成运维中心的各处室部门报表功能的需求，按业务板块建立主报表树，业务部门也可以按所需维度建立特定的逻辑报表树。

统一报表访问平台将逐步实现运维中心报表的统一制作、发布和管理，具体展现方式包括仪表盘、固定报表。

**北京理工大学继续教育暨现代远程教育学院**

**专科学生毕业实践（报告二）**

学生姓名：谢吉星指导教师：姜增如学号：20140111400009 专 业：计算机科学与技术

摘 要

银行数据中心部门较多，日常运维数据相对分散，各部门数据不能实现集中共享，手工报表较多，日常工作量大，且存在同一指标不同来源数据不一致，全中心汇总数据无来源等情况，随着某行业务的逐步扩展，运维中心的规模及数据量级也逐步攀升，对运维的日常工作也随之提高，原有的低效率、高分散性的工作模式已不能实现新一代系统建设的要求，缺乏有效的数据安全管理机制，数据完备性不足，系统应用推广性差、不具备良好可扩展性, 已不能支持更多的新业务应用发展。

因此，数据中心需要建设平稳简单的可扩充性和开放性的运行数据分析平台，实现高质量的数据和统一的数据视图，提供报表、灵活查询功能，最终实现深层数据应用。本论文主要介绍了对运行分析平台的分析、设计和开发的全部过程。运用ER图，程序流程图等对运行分析平台的设计过程进行详细的说明。

全文共分为开发方案、需求分析、系统设计、关键技术解决，结论五部分。开发方案中主要介绍开发运行分析平台的目的、开发方案的选择及开发框架的技术的确定；需求分析介绍了运行分析平台的总体需求及系统各模块的功能需求；系统设计介绍了系统设计的指导思想、数据库的设计、系统模块的设计；关键技术介绍了在具体实现时需解决的一些技术。

本毕业设计的内容是设计并且实现一个银行数据中心运行分析平台，故而系统主要以j2EE作为开发基础,主要使用了Spring+Struts+Mybatis等多种技术，用Eclipse+PLSQL Developer作为开发工具，以ORACLE作为数据库，整合IBM的Cognos报表工具完成报表封装。实现了运行分析平台应有的主要功能模块。该系统界面简单、操作方便，容易维护。

关键词：运行分析平台 j2EE spring +struts +mybatis cognos oracle

**目录**

**[第一章](#_Toc448573026)****[系统概述](#_Toc448573026)** [- 6 -](#_Toc448573026)

[1.1 概述 - 6 -](#_Toc448573027)

[1.2 系统开发的背景 - 6 -](#_Toc448573028)

[1.3 课题研究的意义 - 6 -](#_Toc448573029)

[1.4 系统的开发目标 - 6 -](#_Toc448573030)

**[第二章](#_Toc448573031)****[系统应用的关键技术](#_Toc448573031)** [- 7 -](#_Toc448573031)

[2.1 JQuery技术介绍 - 7 -](#_Toc448573033)

[2.2 关于Struts2 - 8 -](#_Toc448573034)

[2.3 关于Spring MVC - 10 -](#_Toc448573035)

[2.4 关于MyBatis - 12 -](#_Toc448573036)

**[第三章](#_Toc448573038)****[系统需求分析和总体设计](#_Toc448573038)** [- 14 -](#_Toc448573038)

[3.1 设计思想 - 14 -](#_Toc448573043)

[3.2 数据架构设计 - 16 -](#_Toc448573044)

[3.3 技术架构设计 - 18 -](#_Toc448573045)

[3.4 系统功能模块 - 20 -](#_Toc448573046)

[3.5 系统的用例图 - 21 -](#_Toc448573047)

[3.6 实体关系图 - 22 -](#_Toc448573048)

[3.7 数据表的设计 - 23 -](#_Toc448573049)

**[第四章](#_Toc448573051)****[系统的详细设计](#_Toc448573051)** [- 27 -](#_Toc448573051)

[4.1 数据模型设计 - 27 -](#_Toc448573056)

[4.2 ETL设计 - 31 -](#_Toc448573057)

[4.3 主要模块的设计说明与界面 - 36 -](#_Toc448573058)

[4.4 系统的测试 - 58 -](#_Toc448573059)

**[第五章](#_Toc448573061)****[总结与展望](#_Toc448573061)** [- 61 -](#_Toc448573061)

[5.1 工作总结 - 61 -](#_Toc448573064)

[5.2 工作展望 - 63 -](#_Toc448573065)

**[第六章](#_Toc448573066)****[自我鉴定](#_Toc448573066)** [- 64 -](#_Toc448573066)

**[参考文献](#_Toc448573067)** [- 65 -](#_Toc448573067)

**[附录](#_Toc448573068)** [- 66 -](#_Toc448573068)

**[致谢](#_Toc448573069)** [- 69 -](#_Toc448573069)

1. **系统概述**

### 概述

运行数据处理和报告展示，产生了很多业务系统交易日志、事件性能数据、分析诊断信息、配置资产信息、人员机构信息等数据。建设运行分析平台可实现对这些数据的实时、批量统计分析，为集中监控和运维统计分析处理、信息安全和科技风险管理、内外部审计检查等提供支持。

### 系统开发的背景

银行数据中心部门较多，日常运维数据相对分散，各部门数据不能实现集中共享，手工报表较多，日常工作量大，且存在同一指标不同来源数据不一致，全中心汇总数据无来源等情况，随着某行业务的逐步扩展，运维中心的规模及数据量级也逐步攀升，对运维的日常工作也随之提高，原有的低效率、高分散性的工作模式已不能实现新一代系统建设的要求，缺乏有效的数据安全管理机制，数据完备性不足，系统应用推广性差、不具备良好可扩展性, 已不能支持更多的新业务应用发展。

### 课题研究的意义

银行数据中心需要建设平稳简单的可扩充性和开放性的运行数据分析平台，实现高质量的数据和统一的数据视图，提供报表、灵活查询功能，最终实现深层数据应用。

### 系统的开发目标

通过构建统一的业务分析框架，建立全行统一的运行分析环境和基础数据整合平台，来实现对运行中心全方位的管理、分析与决策。

基本实现了运行分析平台应有的主要功能模块,包括：数据模型设计、ETL设计、用户管理、机构管理、菜单管理、角色管理、报表管理、用户登录、用户登出、报表查询等功能。该系统界面简单、操作方便，容易维护。

1. **系统应用的关键技术**

### JQuery技术介绍

JQuery是继prototype之后又一个优秀的Javascript库。它是轻量级的js库 ，它兼容CSS3，还兼容各种浏览器（IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+），jQuery2.0及后续版本将不再支持IE6/7/8浏览器。jQuery使用户能更方便地处理HTML（标准通用标记语言下的一个应用）、events、实现动画效果，并且方便地为网站提供AJAX交互。jQuery还有一个比较大的优势是，它的文档说明很全，而且各种应用也说得很详细，同时还有许多成熟的插件可供选择。jQuery能够使用户的html页面保持代码和html内容分离，也就是说，不用再在html里面插入一堆js来调用命令了，只需要定义id即可。

jQuery是免费、开源的，使用MIT许可协议。jQuery的语法设计可以使开发更加便捷，例如操作文档对象、选择DOM元素、制作动画效果、事件处理、使用Ajax以及其他功能。除此以外，jQuery提供API让开发者编写插件。其模块化的使用方式使开发者可以很轻松的开发出功能强大的静态或动态网页。

jQuery，顾名思义，也就是JavaScript和查询（Query），即是辅助JavaScript开发的库[1]。

### 关于Struts2

#### Struts2简介

Struts 2是Struts的下一代产品。是在 struts 和WebWork的技术基础上进行了合并，全新的Struts 2框架。其全新的Struts 2的体系结构与Struts 1的体系结构的差别巨大。Struts 2以WebWork为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与Servlet API完全脱离开，所以Struts 2可以理解为WebWork的更新产品。因为Struts 2和Struts 1有着太大的变化，但是相对于WebWork，Struts 2只有很小的变化[2]。

#### Struts2架构说明

Struts2框架由三部分构成：核心控制器、业务控制器和用户实现的业务逻辑组件。在这三部分中，struts2框架提供了核心控制器StrutsPrepareAndExecuteFilter，而用户需要实现业务控制层和业务逻辑层。

StrutsPrepareAndExecuteFilter是struts2框架的核心控制器，该控制器作为一个Filter运行在web应用中，它负责拦截所有的用户请求，当用户请求到达时，该Filter会过滤用户请求，如果用户请求以action结尾，该请求就会被转入struts2框架来处理。

Struts2框架获取\*.action请求后，将根据\*.action请求的前面部分决定调用那个业务逻辑组件，Struts2用户处理请求的Action实例，并不是用户实现的业务控制器，而是Action代理--因为用户实现的业务控制器并没有与ServletAPI耦合，显示无法处理用户请求，而struts2框架提供了系列拦截器，该系列拦截器负责将HttpServletRequest请求中的桉树解析出来，传入Action中，并回调Action的execute方法来处理用户请求。

业务控制器组件就是用户Action类的实例，Action类通常包含了一个excute方法，该方法返回一个字符串-----该字符串就是一个逻辑视图名，当业务控制器处理完用户请求后，根据处理结果不同，excute方法返回不同字符串，业务控制器不会对用户请求进行实际处理，用户请求最终由模型组件负责处理，业务控制器只是中间负责调度器，这也是称Action为控制器的原因。

#### Struts2处理流程

Struts2框架的运行流程非常相似于WebWork框架的流程，struts2其实就是WebWork2.2的升级版，因此，struts2的运行流程与WebWork运行流程完全相同，如图WebWork的运行流程：

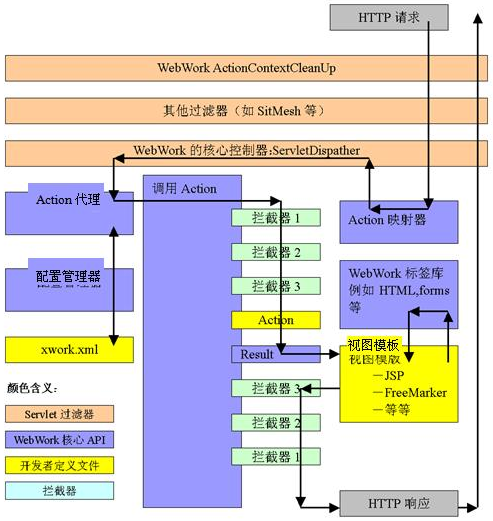


图2-1WebWork运行流程

当用户向Web应用发送请求时，该请求经过ActionContextCleanUp、SiteMesh等过滤器过滤。由WebWork的核心控制器拦截，如果用户请求需要WebWWork的业务逻辑控制器处理，该控制器则调用Action映射器，该映射器将用户请求转发到对应的业务逻辑控制器。值得注意的是，测试的业务逻辑控制器并不是开发者实现的控制器，而是WebWork创建的控制器代理，创建控制器代理时，WebWork需要得到开着定义的xwork,xml配置文件，控制器代理以用户实现的控制器作为目标，以拦截器链中的拦截器作为处理。

### 关于Spring MVC

#### Spring MVC框架介绍

Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。

Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。使用 Spring 可插入的 MVC 架构，可以选择是使用内置的 Spring Web 框架还是 Struts 这样的 Web 框架。通过策略接口，Spring 框架是高度可配置的，而且包含多种视图技术，例如 JavaServer Pages（JSP）技术、Velocity、Tiles、iText 和 POI。Spring MVC 框架并不知道使用的视图，所以不会强迫您只使用 JSP 技术。Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色，这种分离让它们更容易进行定制。

#### Spring MVC 原理图

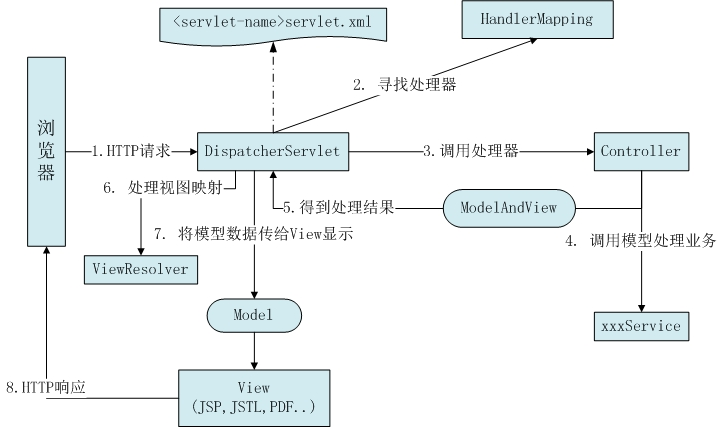


图2-2 SpringMVC原理图

#### Spring MVC 接口解释

DispatcherServlet接口：Spring提供的前端控制器，所有的请求都有经过它来统一分发。在DispatcherServlet将请求分发给Spring Controller之前，需要借助于Spring提供的HandlerMapping定位到具体的Controller。

HandlerMapping接口：能够完成客户请求到Controller映射。

Controller接口：需要为并发用户处理上述请求，因此实现Controller接口时，必须保证线程安全并且可重用。Controller将处理用户请求，这和Struts Action扮演的角色是一致的。一旦Controller处理完用户请求，则返回ModelAndView对象给DispatcherServlet前端控制器，ModelAndView中包含了模型（Model）和视图（View）。

从宏观角度考虑，DispatcherServlet是整个Web应用的控制器；从微观考虑，Controller是单个Http请求处理过程中的控制器，而ModelAndView是Http请求过程中返回的模型（Model）和视图（View）。

ViewResolver接口：Spring提供的视图解析器（ViewResolver）在Web应用中查找View对象，从而将相应结果渲染给客户。

#### Spring MVC 处理流程

客户端请求提交到DispatcherServlet，由DispatcherServlet控制器查询一个或多个HandlerMapping，找到处理请求的Controller，DispatcherServlet将请求提交到Controller，Controller调用业务逻辑处理后，返回ModelAndView

DispatcherServlet查询一个或多个ViewResoler视图解析器，找到ModelAndView指定的视图，视图负责将结果显示到客户端

DispatcherServlet是整个Spring MVC的核心。它负责接收HTTP请求组织协调Spring MVC的各个组成部分。其主要工作有以下三项：截获符合特定格式的URL请求；初始化DispatcherServlet上下文对应的WebApplicationContext，并将其与业务层、持久化层的WebApplicationContext建立关联；初始化Spring MVC的各个组成组件，并装配到DispatcherServlet中[4]。

### 关于MyBatis

#### MyBatis介绍

MyBatis 是支持普通 SQL查询，存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的JDBC代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML或注解用于配置和原始映射，将接口和 Java 的POJOs（Plain Old Java Objects，普通的 Java对象）映射成数据库中的记录。

每个MyBatis应用程序主要都是使用SqlSessionFactory实例的，一个SqlSessionFactory实例可以通过SqlSessionFactoryBuilder获得。SqlSessionFactoryBuilder可以从一个xml配置文件或者一个预定义的配置类的实例获得。

用xml文件构建SqlSessionFactory实例是非常简单的事情。推荐在这个配置中使用类路径资源（classpath resource)，但你可以使用任何Reader实例，包括用文件路径或file://开头的url创建的实例。MyBatis有一个实用类Resources，它有很多方法，可以方便地从类路径及其它位置加载资源。

#### MyBatis框架架构

配置来源于两个地方，一处是配置文件，一处是Java代码的注解，将SQL的配置信息加载成为一个个MappedStatement对象（包括了传入参数映射配置、执行的SQL语句、结果映射配置），存储在内存中。

SQL解析：当API接口层接收到调用请求时，会接收到传入SQL的ID和传入对象（可以是Map、JavaBean或者基本数据类型），Mybatis会根据SQL的ID找到对应的MappedStatement，然后根据传入参数对象对MappedStatement进行解析，解析后可以得到最终要执行的SQL语句和参数。

SQL执行：将最终得到的SQL和参数拿到数据库进行执行，得到操作数据库的结果。

结果映射：将操作数据库的结果按照映射的配置进行转换，可以转换成HashMap、JavaBean或者基本数据类型，并将最终结果返回。

### 小结

本章全面介绍本系统所用到的关键技术和工具，其中JQuery是继prototype之后又一个优秀的Javascript库，它是轻量级的js库，它兼容CSS3，还兼容各种浏览器，使用户能更方便地处理Html、Events、实现动画效果，并且方便地为网站提供Ajax交互，SSM(struts+spring+mybatis)作为本系统的核心与亮点技术[5]，更是作了详细介绍，要掌握他们的工作原理，主要功能。掌握了这些技术基础，有利于理解本系统的功能实现的处理逻辑。

1. **系统需求分析和总体设计**

### 设计思想

#### 技术路线

运行分析平台设计时将遵循以下技术路线：

* 运行分析平台总体设计符合SOA整体架构规范，按照银行的IT总体规划和银行系统建设目标要求，准确定位运行分析平台及相关的各个系统的功能及服务内容，系统间遵循标准化服务调用模式。
* 运行分析平台总体按照最优投资的建设思想，充分利用行内现有的软件资源，如操作系统、数据库系统、中间件系统和流程系统软件。
* 运行分析平台总体按照标准化的方式符合全行的流程管理规范和技术开发规范，达到行内系统应用的规范化和技术统一化；与行内系统使用相同的操作系统（RedHatLinux）、数据库系统（Oracle）、BI分析工具（Cognos）、ETL工具等。

#### 设计思想

* **先进性原则**

运行分析平台设计体现优化系统结构、完善系统功能、强化处理能力、简化操作流程的原则；

运行分析平台支持主流通信协议和规范，系统在架构上采用基于框架和组件，层次化的架构设计方式，前端采用Web界面，整体架构符合J2EE规范，同时支持SOA架构。

* **前瞻性原则**

运行分析平台采用当前成熟的技术，符合国际、国内标准的软硬件技术规范，软件设计思想成熟稳定；

运行分析平台总体架构和软件体系结构具有充分的前瞻性，充分考虑未来业务发展和管理的变化，方便对新业务和新需求的扩展和支持。

* **可扩展性原则**

运行分析平台所选用的系统软硬件平台具有良好的可扩充能力，支持系统规模的扩大和业务范围的扩展以及设计容量的扩展，能够满足今后业务发展的需要。能够在不更改系统的软件结构和网络结构的前提下，方便的支持系统扩容；

为适应将来业务和技术发展的需求，运行分析平台在设计上采用模块化和松耦合的思路，具有较强的独立性和高度的可扩展性。该产品具有易于维护的特点，除具有标准的开放式技术接口外，还能够完成与现有系统具有标准接口的系统完全对接；

运行分析平台采用分布式部署方式，采用数据与应用分离的方式，以及数据库操作接口封装服务的机制支持多数据源访问连接，使得系统间交互与集成更加方便；

### 数据架构设计

#### 数据模型框架

系统数据架构自下而上分为贴源层、基础数据层、汇总数据层、应用数据层四个部分。

贴源层的数据采取与源系统数据同构的方式；

在基础数据层开始将各个源系统的数据进行统一标准化，按分析主题进行划分，归并来自各个系统的同类数据；

在汇总数据层对数据进行分类汇总，建立多维数据立方体；

在应用数据层建立贴近应用的数据集。



图3-1系统数据架构

#### 数据标准

* **贴源层数据**

对于贴源层的数据表，采用同名同构的设计方式，即：贴源层的表的结构与源系统的表的结构相同，有利于数据装载；贴源层的表的名称采用源系统的表的名称，为了区别表的系统来源及避免不同系统的同名表冲突，在表名称前增加系统的代码前缀。例如：来自CMDB系统（系统的代码前缀为S01\_）的表BMC\_CORE\_BMC\_Application，在贴源层中的名称为S01\_BMC\_CORE\_BMC\_Application。

各源系统的代码前缀如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **代码前缀** | **系统缩写** | **系统名称** | **备注** |
| S01 | CMDB | 配置管理数据库 |  |
| S02 | ITSM | 服务流程 |  |
| S03 | EM | 环境部 |  |
| S04 | TPMS | 交易量集中监控 |  |
| S05 | SRM | 存储 |  |
| S06 | IOMP | 云管理平台 |  |
| S07 | CMMP | 集中监控 |  |

表3-1 源系统代码前缀

* **基础数据层数据**

基础数据层的数据表名称以“TXX\_”开头，其中“XX”是主题编号标识，后面用于标识实体的英文名称组成数据表名。

各个主题的缩写代码约定如下：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **缩写代码** | **主题代码** | **主题名称** |
| 01 | T01 | Party | 当事人 |
| 02 | T02 | Infrastructure | 基础软件组件 |
| 03 | T03 | Device | 设备设施 |
| 04 | T04 | Event | 事件 |
| 05 | T05 | Business Unit | 业务单元 |
| 06 | T06 | Address | 地址 |

表3-2 主题代码约定

从基础汇数据层开始，源系统的各个码表的代码，将在该层开始进行统一。不一致的，需要进行转换。

* **汇总数据层数据**

汇总数据层的表以“TB\_SUM\_”开头；

* **应用数据层数据**

应用数据层的表以“TB\_APP\_”开头；

### 技术架构设计

运行分析平台技术方面主要包含ETL、任务调度、数据库、逻辑对象、业务对象、业务控制层、表现层等部分组成，这几部分相对独立，分别基于不同的产品实现。

图3-2技术架构

系统主要分为5层：表现层、业务控制层、业务对象层、逻辑对象层、数据层和2个环节：作业调度和ETL处理。每个层之间相互独立，并且是松耦合，互相影响比较小。这样的结构保证系统结构清晰、开发灵活、扩展方便。上层都可以方便的调用下层的接口，但不能跨层调用，保证了系统的稳定性。

（1）表现层定义输入、输出、显示界面。如表单、列表、菜单等。使用JavaScript、PDM Tag（标签库）、Dynamic HTMPL（动态脚本语言）技术实现。

（2）业务控制层实现业务逻辑，如事件查询、流转、编辑、关闭等业务模块。主要使用的技术有Workflow Engine（工作流程引擎）、Action Command Center（业务命令处理器）、Web Service。

（3）业务对象层主要实现企业对象的管理，企业对象是定义业务系统某快业务对象，如事件对象，还可以为这些对象设计属性、方法和函数。使用的技术包括Business Object（商业对象技术）、Object Cache Pool（对象缓冲池管理技术，将企业对象在缓冲池内管理，减少数据库连接，极大提高了系统的性能）、Object Design（企业对象设计器）。

（4）逻辑对象层定义与数据库表相关的逻辑对象，实现对数据结构统一定义与管理（不必关心是什么类型的数据库，如SQL Server和Oracle）。主要技术有Logic Object（逻辑对象定义）、Logic Interface（逻辑对象接口技术

（5）数据层是系统的数据存储。本系统采用Oracle 11g来持久存储及管理数据。

（6）作业调度是日终批量作业的集中调度环节。本系统采用完全由开源作业调度平台TaskCTL进行作业统一调度。

（7）ETL是系统对数据的抽取、转换、装载的操作，采用Oracle SQLLOADER实现数据的装载，使用ORACLE的存储过程实现数据的清洗、加工和转换已经数据质量的检查；

### 系统功能模块

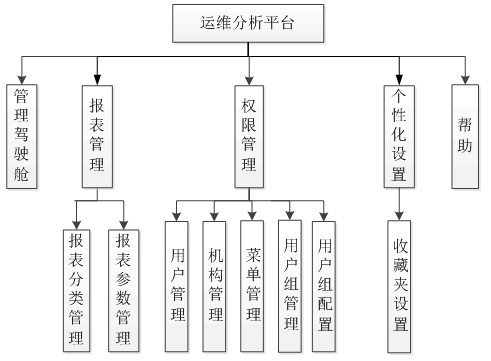


图3-3系统功能模块

在本系统中主要阐述报表管理、系统管理、个性化设置等模块。

（1）报表管理：实现报表搜索，报表分类。

（2）系统管理：实现用户管理、机构管理、权限管理、用户组管理、用户组配置等。

（3）个性化设置：收藏夹设置。

### 系统的用例图

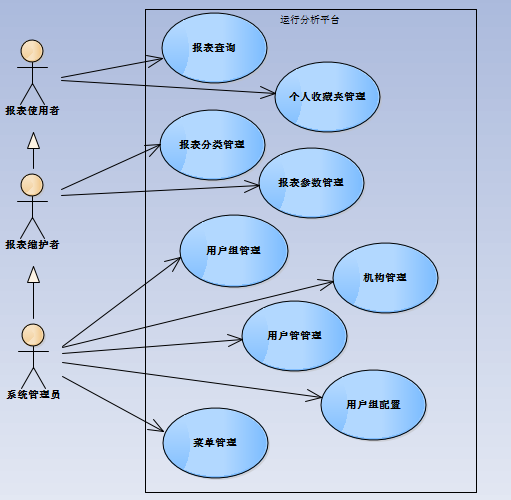


图3-4总体用例图

如图3-9，报表查询、个性化设置、报表分类管理、报表参数管理、用户管理、机构管理、用户组管理、用户组配置、菜单管理等模块组成。使用者是指报表使用者、报表维护者、系统管理员。

(1)报表查看人员

用户有权限查看所属部门的指定报表，修改个人的信息的功能。

(2)报表开发人员

用户可以使用的报表设计器进行报表的设计和开发。

(3)系统管理员

用户管理、机构管理、用户组管理、用户组配置、菜单管理等功能。

### 实体关系图

下面是该系统数据库表的E-R图，如图3-5

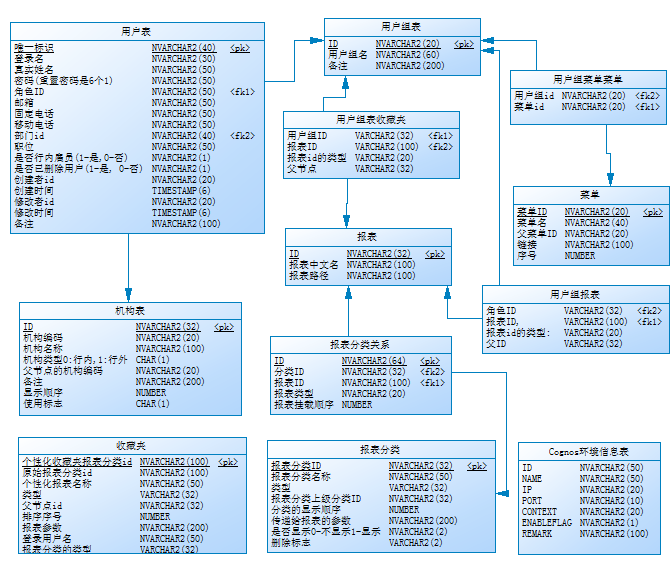


图3-5 E-R图

### 数据表的设计

下面列出了设计过程中所用到的数据库表及其结构：

1.机构表OAP\_TB\_ORG 如表3-3：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 32 | 流水号 |
| ORG\_CD | NVARCHAR2 | 20 | 机构编码 |
| NAME | NVARCHAR2 | 100 | 机构名称 |
| TYPE | CHAR | 1 | 机构类型0:行内,1:行外 |
| PCD | NVARCHAR2 | 20 | 父机构编码 |
| REMARK | NVARCHAR2 | 200 | 备注 |
| ORDER\_NUM | NUMBER | 4 | 显示顺序 |
| USE\_FLG | CHAR | 1 | 使用标志1:启用,0：停用 |

表3-3 机构表

2.用户表OAP\_TB\_USER，如表3-4：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 40 | 唯一标识 |
| LOGIN\_NAME | NVARCHAR2 | 30 | 登录名 |
| NATURE\_NAME | NVARCHAR2 | 50 | 真实姓名 |
| PASSWORD | NVARCHAR2 | 50 | 密码 |
| RID | NVARCHAR2 | 50 | 角色ID |
| EMAIL | NVARCHAR2 | 50 | 邮箱 |
| TEL | NVARCHAR2 | 50 | 固定电话 |
| MOBILE | NVARCHAR2 | 50 | 移动电话 |
| DEPART\_ID | NVARCHAR2 | 40 | 部门ID |
| POSITION | NVARCHAR2 | 50 | 职位 |
| IS\_EMP | NVARCHAR2 | 1 | 是否行内雇员1-是,0-否 |
| IS\_DEL | NVARCHAR2 | 1 | 是否删除1-是, 0-否 |
| CREATER | NVARCHAR2 | 20 | 创建者ID |
| CREATE\_TIME | TIMESTAMP |  | 创建时间 |
| MODIFIER | NVARCHAR2 | 20 | 修改者ID |
| MODIFY\_TIME | TIMESTAMP |  | 修改时间 |
| REMARK | NVARCHAR2 | 100 | 备注 |

表3-4 用户表

3.用户组权限表OAP\_TB\_PRIVILEGE\_USERGROUP如表3-5：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| GROUP\_ID | VARCHAR2 | 32 | 用户组ID |
| PRIVILEGE\_ID | VARCHAR2 | 64 | 权限ID |
| TYPE | VARCHAR2 | 1 | 用户组类型：  type=0, 权限ID表示菜单ID, type=1, 权限ID表示报表分类ID,  type=2, 权限Id表示报表分类与报表关系ID |

表3-5 用户组权限

4.用户组表OAP\_TB\_USERGROUP如表3-6：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| GROUP\_ID | VARCHAR2 | 32 | 用户组id |
| USER\_ID | VARCHAR2 | 32 | 用户id |
| TYPE | VARCHAR2 | 1 | 关系类型 |

表3-6用户组

5.菜单表OAP\_TB\_ROLE如表3-7：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 20 | 菜单ID |
| NAME | NVARCHAR2 | 40 | 菜单名 |
| PID | NVARCHAR2 | 20 | 父菜单ID |
| MENU\_URL | NVARCHAR2 | 100 | 链接 |
| MENU\_ORDER | NUMBER | 4 | 序号 |

表3-7 菜单表

6.报表配置表OAP\_TB\_REPORT如表3-8：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 32 | 流水号 |
| NAME | NVARCHAR2 | 100 | 报表中文名 |
| URL | NVARCHAR2 | 100 | 报表路径 |

表3-8：报表配置表

7.用户组报表收藏夹OAP\_TB\_ROLE\_REPORT\_FAVOR 如表3-9：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ROLE\_ID\_FAVOR | VARCHAR2 | 32 | 用户组ID |
| REPORT\_ID\_FAVOR | VARCHAR2 | 100 | 报表ID |
| TYPE | VARCHAR2 | 20 | 报表id的类型 |
| PID | VARCHAR2 | 32 | 父节点：当type=1时, pid指向reportId的父节点 |

表3-9：用户组报表收藏夹

8.报表分类关系表 OAP\_TB\_REPORT\_RPTCATEGORY如表3-10：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 64 | ID |
| CID | NVARCHAR2 | 32 | 分类ID |
| RID | NVARCHAR2 | 100 | 报表ID |
| RTYPE | NVARCHAR2 | 20 | 报表类型 |
| ORDER\_NUM | NUMBER | 4 | 报表挂载顺序 |

表3-10：报表分类关系表

9.报表分类表 OAP\_TB\_RPTCATEGORY如表3-11：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 32 | 报表分类ID |
| NAME | NVARCHAR2 | 50 | 报表分类名称 |
| TYPE | VARCHAR2 | 32 | 类型 0:个人收藏夹;1:按部室分类;2:按系统分类 |
| PID | NVARCHAR2 | 32 | 报表分类上级分类ID |
| ORDER\_NUM | NUMBER | 4 | 分类的显示顺序 |
| PARAM | NVARCHAR2 | 200 | 传递给报表的参数 |
| STATUS | NVARCHAR2 | 2 | 是否显示：0-不显示1-显示 |
| DELETE\_SIGN | VARCHAR2 | 2 | 删除标志 |

表3-11：报表分类表

10.收藏夹表 OAP\_TB\_FAVORITE 如表3-12：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 长度 | 描述 |
| ID | NVARCHAR2 | 100 | 个性化收藏夹报表分类id |
| REPORTID | NVARCHAR2 | 100 | 原始报表分类id |
| NAME | NVARCHAR2 | 50 | 个性化报表名称 |
| FAVORTYPE | VARCHAR2 | 32 | 类型0,文件夹,1,报表 |
| PID | NVARCHAR2 | 32 | 父节点id |
| ORDERNUM | NUMBER | 4 | 排序序号 |
| PARAM | NVARCHAR2 | 200 | 报表参数 |
| USERNAME | NVARCHAR2 | 50 | 登录用户名 |
| TYPE | VARCHAR2 | 32 | 报表分类的类型 |

表3-12：收藏夹表

### 小结

本章首先分析系统的功能需求，通过对系统功能模块的划分以及角色的划分，明确了系统需求，为总体设计提供依据。接下来阐述了系统的总体设计思想，说明了本系统是基于B/S架构下的多层结构应用系统。通过数据库的设计分析，以及各个模块用例图的描述，形成了开发的整体思路，为下一步编程实现各个具体功能铺平了道路。

1. **系统的详细设计**

### 数据模型设计



#### 源数据分析

数据源分析的主要作用是了解业务系统数据的情况，为建立贴源层、基础层、分析层、汇总层以及数据加载做准备的。分析的内容主要有：数据表分析、表字段分析、代码分析等。

有了源数据分析的结果，我们就可以结合业务模型来制定平台中各存储部分的数据标准，方便今后进行数据整合的工作。

#### 贴源层存储设计

贴源数据存储的设计比较简单，主要是在源数据分析的基础上，与各业务系统确定好数据接口和供数规则和供数周期的基础上，在平台中设计相应的数据库表，装载各业务系统提供的接口数据，便于把后续接口数据整合到平台中。

该层数据也即平台的源数据缓冲区，主要是为了便于平台各数据层后续的ETL工作，和便于后续数据检查和核对工作，所以在进行贴源层数据存储设计的时候需要注意以下几点：

* 数据面尽可能的涵盖目前所有的原业务系统。
* 保持所有原业务系统的数据原貌。
* 按照业务系统的不同进行数据的分表分区来处理，便于后续数据的维护。

#### 基础层存储设计

* T01当事人

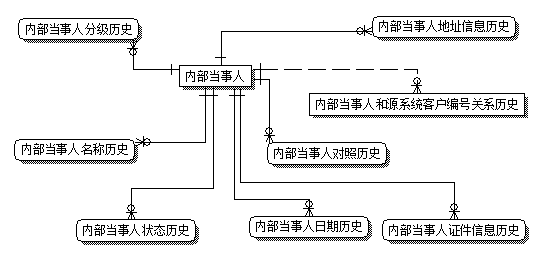


图4-1当事人

* T02基础软件组件

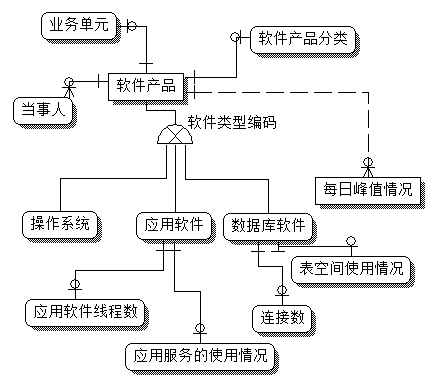


图4-2 基础软件组件

* T03设备设施组件

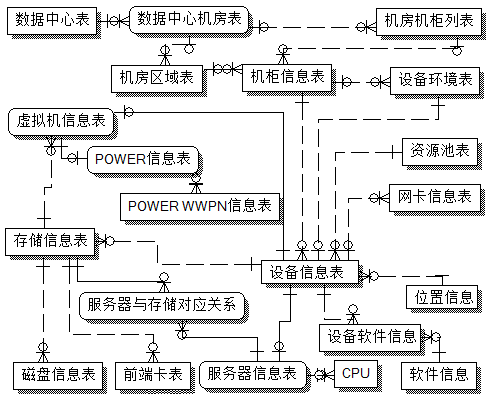


图4-3 设备设施组件

* T04事件

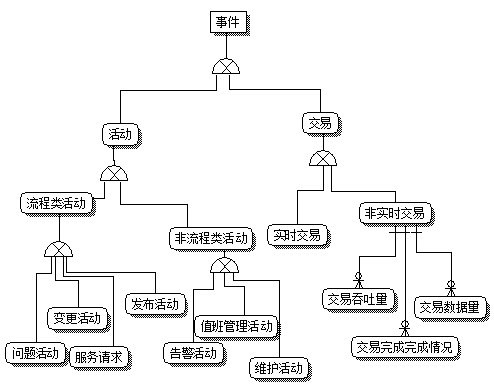


图4-4事件

* T05业务单元

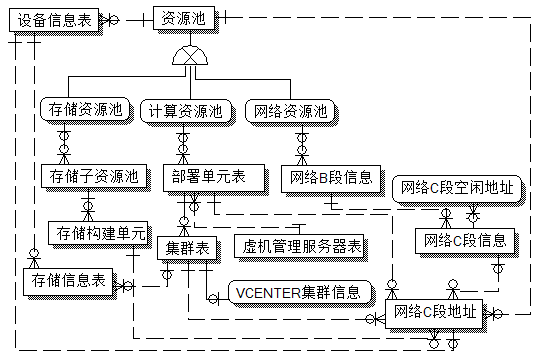


图4-5 业务单元

* T06 位置地址

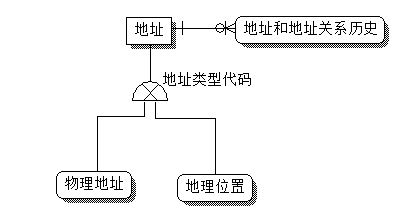


图4-6位置地址

#### 汇总层存储设计

汇总层存储的设计主要是对进行周期汇总统计的指标信息管理，针对同一个指标，该层一般有相应的日统计和月统计表，日统计表往往都是为月统计表提供数据来源，所以日统计表一般都只保留最近的32天的数据，月统计表一般保留近三年的月统计信息和之前每年最后一个月的月统计信息。这样对于周期类的指标统计需求，该层数据一般都能支持，对于日统计中的最近32天前的数据，无需保留，因为主要的指标信息基本上在汇总层的历史表中都有记录下来，所以历史上的每一天的信息都能查到。业务应用中比较常见的月末时点指标数据往往在ADM层的月统计信息中都有月末时点的指标记录[6]。

在进行ADM存储设计的时候需要注意以下几点：充分分析业务应用需求的共性，提炼常用的业务指标；对指标纬度进行最精细化的设计，从而达到不随业务需求的变动而变更指标信息；定期的数据平衡检查，保证了数据的完整性和准确性；结合数据类型的不同、数据逻辑主题的不同进行数据存储策略的调整[7]

### ETL设计

ETL是将数据源的数据经过一系列转换处理，加载到本系统之中，也就是数据抽取、数据转换、数据传输、数据加载的过程：

图4-7ETL转换处理过程

银行运行分析平台项目的ETL过程包括以下逻辑部分：

* 数据源
* 数据抽取
* 数据接口
* 数据清洗
* 数据加载
* 数据转换

每一逻辑单元还包括运行在一系列硬件和软件平台上的物理过程。整体ETL过程如下：

* 源系统推送的数据文件到运行分析平台ETL服务器上；
* 进行必要的文件校验，和基本的数据清理过程，本阶段的清理过程最主要的目标是为了保证数据能够快速无误地将数据加载到Oracle数据库中；
* 对处理后的文本文件利用SQLLOADER将数据加载到数据缓冲层；
* 对缓冲层的数据同基础层的全量历史数据进行增量比对，并按照标准化过程将数据加工到基础层；
* 按照最终的业务应用需求，将数据进行必要的加工并形成最终的应用数据；

#### 数据源

数据源采集范围列表：网络管理系统、云管理、新一代集中监控、大机监控；

#### 数据接口

* **文件命名规范**

每个表卸载成一个文件，文件名称为大写表名称加扩展名.dat（适用于数据库卸载方式）或者.csv(标准csv格式，适用于日志卸载方式)作为数据文件名称，如：表T\_ACALARM卸载的文件命名为：T\_ACALARM.dat；

每个系统卸载的所有数据文件，以及一个包含该系统所有数据文件校验信息的校验文件，打包压缩，打包压缩的文件格式为SYSNAME\_yyyymmdd.tar.gz。校验文件SYSNAME.verf。

该校验文件格式如下：数据文件名、表数据行数，以空格分隔，每个表信息占一行。例如：T\_ACALARM.dat 100

打包采用tar命令，压缩采用tar本身的z参数，或系统的gzip命令进行压缩，该压缩文件解压后不带目录。

打包文件上传完成后，再上传一个SYSNAME\_yyyymmdd.ok的文件。该校验文件格式如下：压缩包字节大小，压缩包的md5校验码，以空格分隔；

例如：28ed8215a6491057e41ed7e00d479abf1b

* **数据文件格式规范**

规范一（适用于数据库卸载）

在每个数据文件中，数据格式如下：

每个文件的第一行是列名称，数据文件行结束符用0x01，数据文件列分隔符用逗号（注：每列后面带一个分隔符）；所有字段值用单字节0x02包含起来；值为空的列，卸载成空串，不要卸载成NULL；源系统供数sql语句，写成select 列名，列名按照建表字段顺序安排，采用GBK字符集编码，日期格式用yyyy-mm-dd格式：日期时间格式为yyyy-mm-dd hh:mm:ss，含有毫秒的时间格式为yyyy-mm-dd hh:mm:ss.xxxxxx；

规范二（标准csv格式，适用于日志卸载）

在每个数据文件中，数据格式如下：

每个文件的第一行是列名称，数据文件行结束符用换行符（0xa），数据文件列分隔符用逗号，非数值用单字节双引号包含起来，值为空的列，卸载成空串，不要卸载成NULL；源系统供数sql语句，写成select 列名，列名按照建表字段顺序安排，采用GBK字符集编码，日期格式用yyyy-mm-dd格式：日期时间格式为yyyy-mm-dd hh:mm:ss，含有毫秒的时间格式为yyyy-mm-dd hh:mm:ss.xxxxxx；

#### 数据抽取

数据抽取指从源业务系统按照运行分析平台数据接口的规范要求，抽取运行分析平台需要的各种数据，形成符合标准接口的数据文件。

数据抽取策略主要分两类：全量抽取、增量抽取。全量抽取是将数据表从数据源系统中全部抽取出来；增量抽取是仅抽取上次抽取以后变化的数据。增量抽取的优点是抽取的数据量小，从而转换和加载的数据量也小，能够极大提高数据加载性能。对于大数据量，建议采用增量抽取；仅当数据量很小，或实在无法分离出增量数据时，才采用全量抽取。

数据抽取对各业务系统的依赖性强，应该根据各业务系统实际情况制定相应的抽取策略，在项目实施时考虑，包括抽取方式、抽取时间、抽取周期等内容[6]。

#### 数据清洗

对抽取待加载的业务数据，可能需要进行初步的数据清洗工作，其目的是为了避免在数据加载过程中由于数据质量问题而导致数据加载过程失败。所以清洗的最主要目标：保证加载工具能够快速无异常地进行数据加载。对于可能导致加载过程失败的内容进行转换和清洗。

项目中我们使用Oracle存储过程实现对数据的加工清洗；

#### 数据加载

广义上的数据加载过程包括了：从接口文件加载到数据库中的数据缓冲层以及按照某种加载策略，把缓冲层的数据加载到基础层。

对于运行分析平台来说，加载任务包括两种，其一是初始加载，也就是在系统正式运行之前向系统中装入历史数据；其次是日常加载，它使运行分析平台数据和源系统的数据保持一致。初始加载的工作是一次性的过程，而日常加载是系统正常运行过程中进行日常数据处理的过程。在设计过程中，需要对日常加载任务进行更为细致地分析和设计。

从源系统推送的文件经过解压缩，校验、清洗之后加载进入oracle数据缓冲层。

缓冲层到基础层的数据处理过程要遵循一定的数据加载策略。数据加载策略主要指数据加载的方式，这可以根据数据的抽取策略以及业务规则来确定，一般有三种类型：全表覆盖、记录追加、历史更新。ETL任务会根据源系统提供的接口数据情况来决定具体的加载策略，原则上业务系统能获取增量的尽量以增量的方式提供，便于提高数据处理效率，压缩ETL处理的时间窗口，从而满足对外提供数据的需求[6]；

本项目利用Oracle强大的并行数据处理能力，把对数据的转换工作也归入到加载过程来描述工作放在数据库内实现。

#### 数据转换

从缓冲层到基础层的数据转换基本有三种：包括代码转换的数据基本格式转换；从多个数据项中生成派生数据，例如，如果A并且C，就生成D；通过从多个数据源进行数据加工合并生成数据。

转换的规则在数据定义、Mapping过程中必须有清晰的描述。

数据转换主要需要考虑的是效率和维护工作量，效率方面主要是力求避免数据转换成为ETL过程的瓶颈，由于I/O资源耗费大，因此全部记录级转换应该一次完成，避免对源表的多次扫描。对于有公性的转换，比如公共代码的转换，尽量采用标准的处理函数来实现。

从基础层到汇总层，加汇总到应用层的数据转换，则大多数是属于遵守业务要求的数据汇总和转换处理。

#### 数据归档

源数据文件通过ETL工具加载到数据库之后，被压缩转移到归档目录。

#### 任务调度与监控

业务系统的数据源众多，源系统、数据间逻辑关系相当复杂，为了保证逻辑清晰，必须将ETL任务归类划分，并根据ETL任务的特点及相互间的依赖和触发关系规划出各类ETL任务运行的方式、方法，来保证各类数据能够安全、有效、快速地加载到运行平台。

本系统的调度任务采用的是开源的TASKCTL工具。

TASKCTL是一款先进的批量任务控制技术平台，可以为各行业诸多应用场景服务，并提供多种应用解决方案。利用TASKCTL技术平台，可以实现整个企业的统一调度，统一驱动，用户可以打造强大的ETL批量调度平台，灵活的系统级、业务级监控平台，用户还可以利用TASKCTL的灵活的流程自动化特征，实现自己的一些自动化流程处理。

TASKCTL不仅可以为企业提供全面优秀的批量调度解决方案，也可以为企业提供灵活的各种系统级、应用级监控方案；同时，TASKCTL还可以为企业运维提供各种自动化解决方案。TASKCTL不仅可以作为企业基础设施，也可作为运维管理的灵活工具。

可以实现作业、作业流之间的并行、互斥以及任意依赖控制。特别是依赖控制，系统通过串行、单点依赖、事件依赖以及自定义条件等机制，可以实现作业流内、不同作业流、不同ETL作业服务器以及不同业务日期、不同批次间任意作业的依赖控制。

为了用户适时了解作业的运行状况，TASKCTL整体采用实时刷新、图形、多角度多口径统计以及短信等方式对整个平台作业进行全方位监控，以便用户及时哪些作业正在运行、哪些作业错误以及错误原因等信息。

### 主要模块的设计说明与界面



#### 登录管理

用户登录模块主要包括用户登陆、修改密码功能；

##### 用户登录功能

此功能主要是根据用户输入的用户名、密码来完成用户的身份识别以及根据身份信息组织主页面菜单的功能。

* 处理逻辑

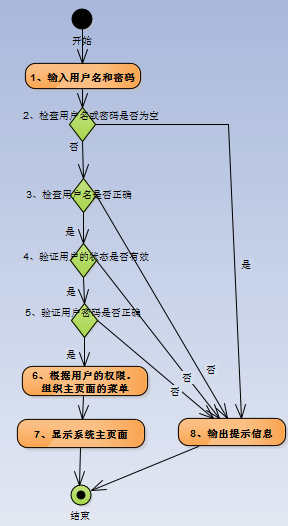


图4-8用户登录流程图

表4-1用户登录流程说明

|  |  |  |
| --- | --- | --- |
| 序号 | 流程描述 | 说明 |
| 1 | 输入用户名和密码 |  |
| 2 | 检查用户名或密码是否为空 |  |
| 3 | 检查用户名是否正确 |  |
| 4 | 验证用户的状态是否有效 |  |
| 5 | 验证用户密码是否正确 |  |
| 6 | 根据用户的权限，组织主页面的菜单 |  |
| 7 | 显示系统主页面 |  |
| 8 | 输出提示信息 | 2.1用户名或密码不能为空；  3.1用户名输入错误；  4.1用户名不存在；  4.2用户已经被锁定；  5.1输入的密码有误； |

* 类图

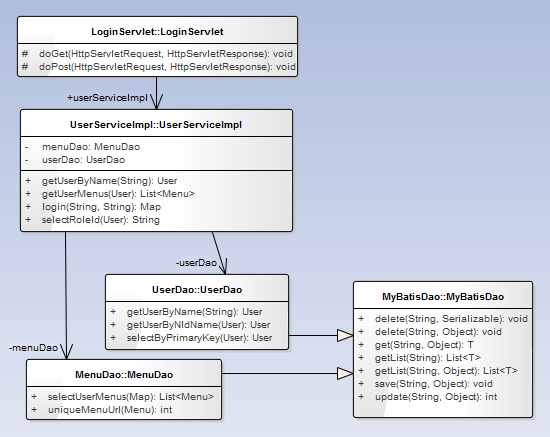


图4-9用户登录类图

* 页面原型



图4-10用户登陆页面

用户登录成功后的系统主页面如图所示：



图4-11系统管理员主界面

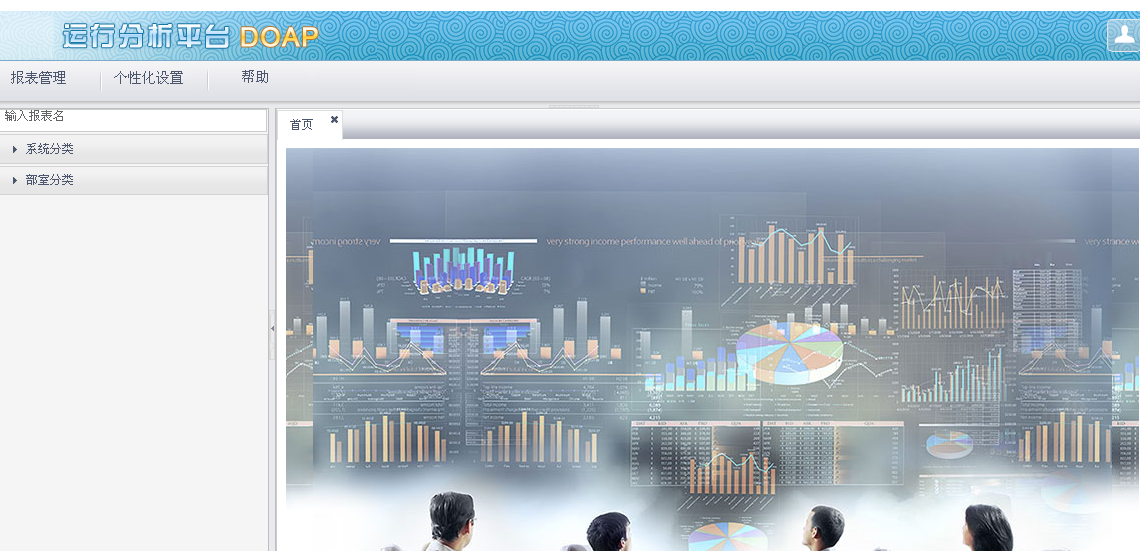


图4-12报表维护员主界面



图4-13报表查询人员主界面

##### 用户修改密码模块

* 程序入口文件

/doap/WebContent/index.jsp

* 处理流程

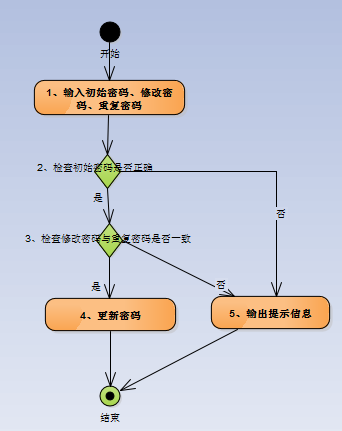


图4-14 用户修改密码流程图

表4-2用户修改密码流程

|  |  |  |
| --- | --- | --- |
| 序号 | 流程描述 | 说明 |
| 1 | 输入初始密码、修改密码、重复密码 |  |
| 2 | 检查初始密码是否正确 |  |
| 3 | 检查修改密码与重复密码是否一致 |  |
| 4 | 更新密码 |  |
| 5 | 输出提示信息 | 2.1初始化输入错误；  3.1两次输入的密码不一致； |

* 类图

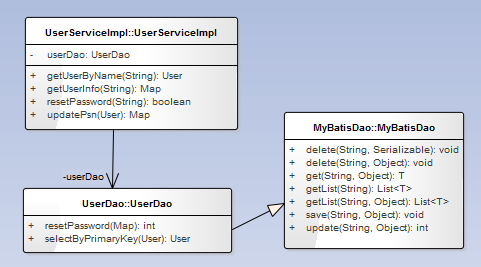


图4-15 用户修改密码类图

* 页面原型

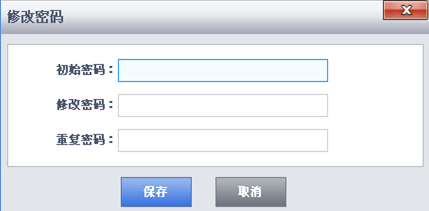


图4-16用户修改密码页面

#### 管理驾驶舱

##### 变更日历

用户以系统管理员身份登录系统，点击“管理驾驶舱”菜单中“变更日历”，打开变更日历页面：

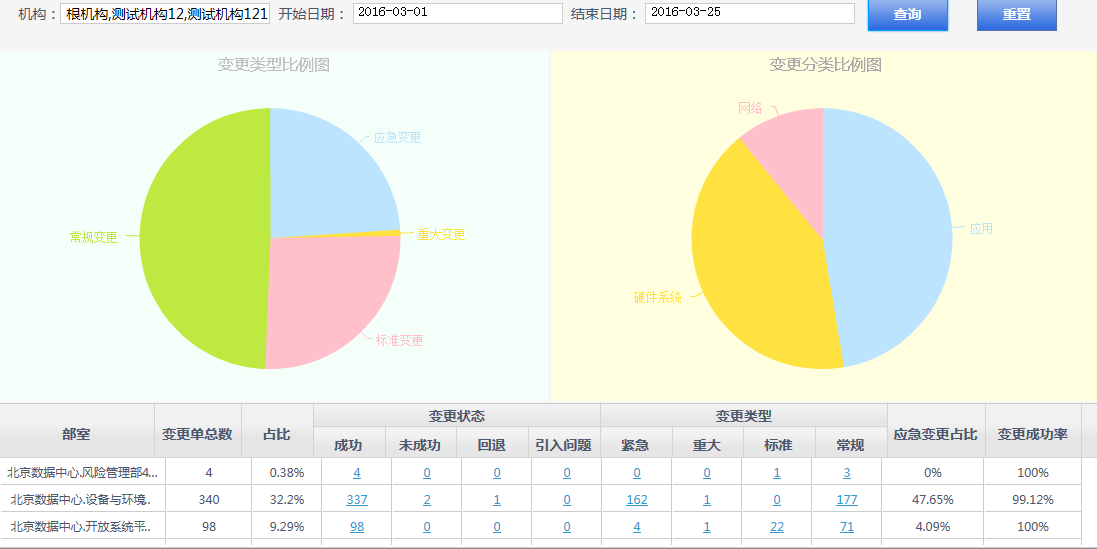


图4-17变更日历页面

##### 交易量分析

用户以系统管理员身份登录系统，点击“管理驾驶舱”菜单中“交易量分析”，打开交易量分析页面；

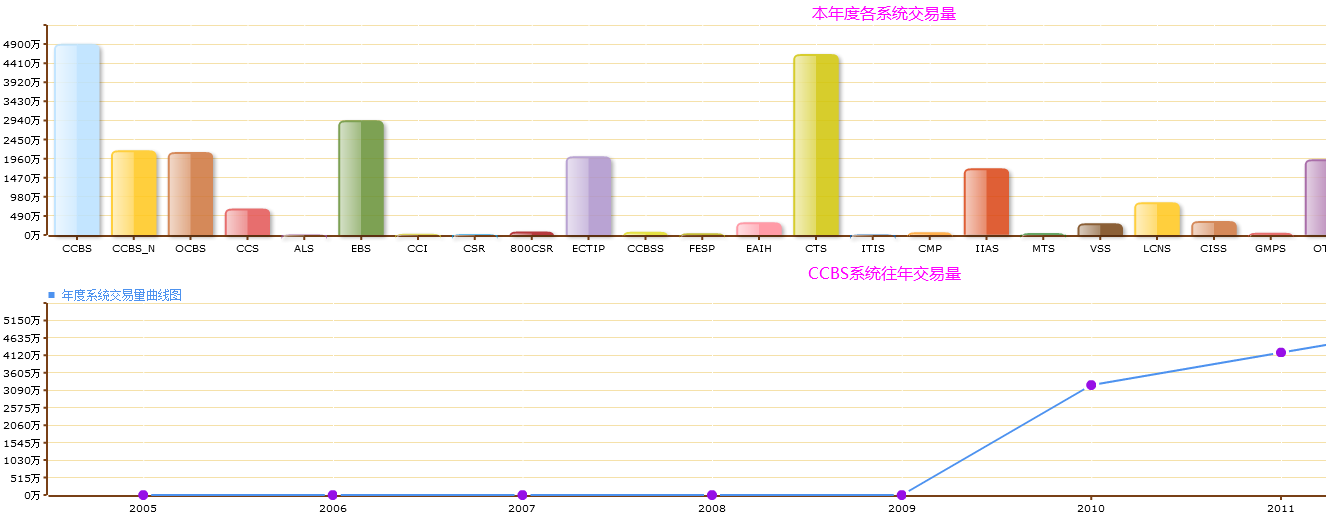


图4-18交易量分析页面

##### 事件分析

用户以系统管理员身份登录系统，点击“管理驾驶舱”菜单中“事件分析”，打开事件分析页面；

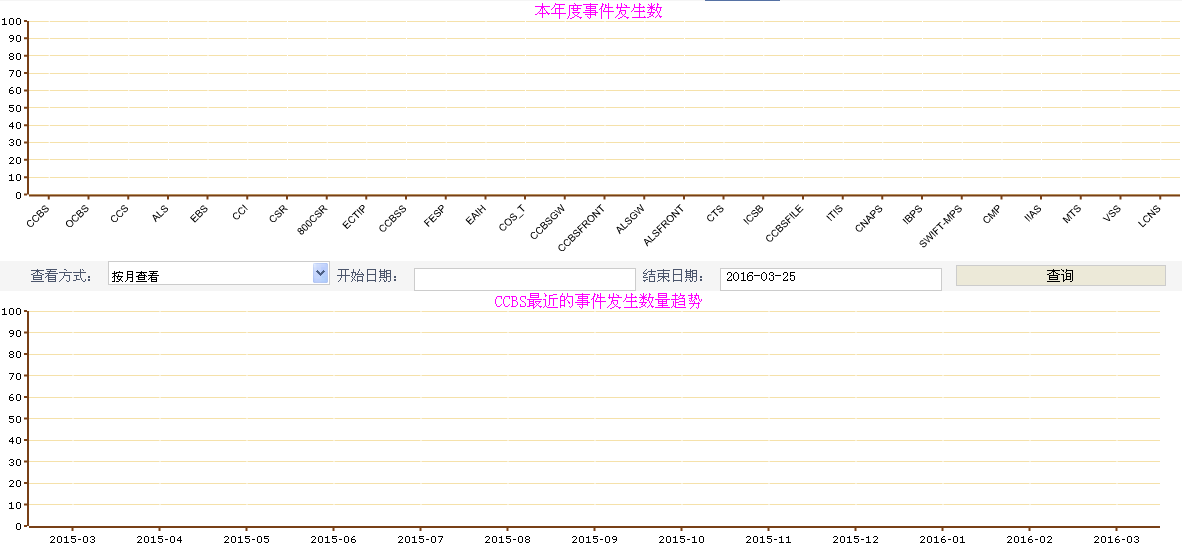


图4-19事件分析页面

##### 系统可用量分析

用户以系统管理员身份登录系统，点击“管理驾驶舱”菜单中“系统可用量分析”，打开系统可用量分析页面；

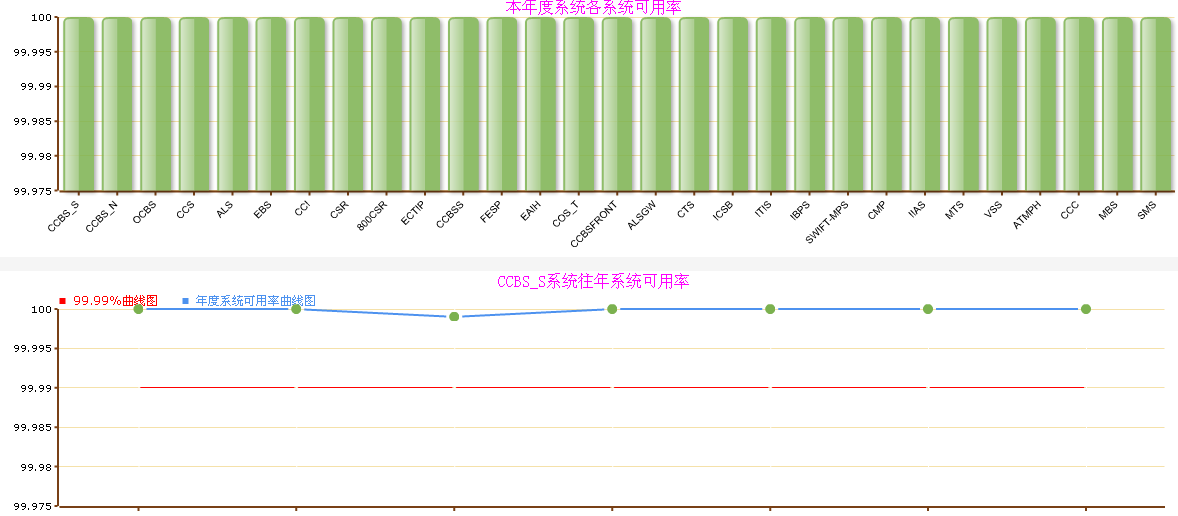


图4-20系统可用量分析页面

#### 权限管理

权限管理包括菜单管理、机构管理、用户管理、用户组管理功能；

##### 菜单管理

用户以系统管理员身份登录系统，点击“权限管理”菜单中“菜单管理”，打开菜单管理页面；

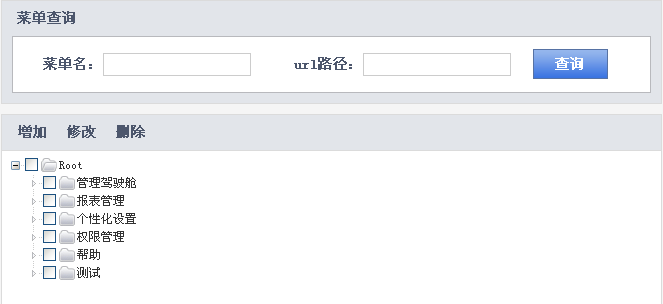


图4-21 菜单管理页面

页面中：“菜单名”是查询条件；

“url路径”是查询条件；

* 增加菜单

点击“增加”，出现增加菜单。

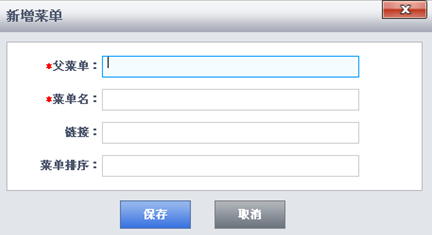


图4-22 新增菜单页面

页面中：“父菜单”为必填项；

“菜单名”为必填项；

点击“保存”，完成新增。

* 修改菜单

选择一个需要修改的菜单，点击“修改”，出现修改菜单页面；

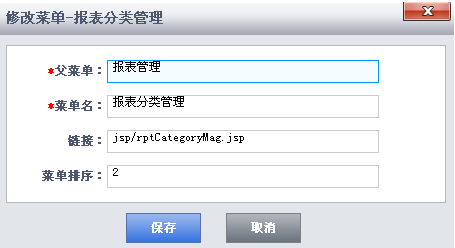


图4-23 修改菜单页面

页面中：“父菜单”为必填项，

“菜单名”为必填项；

点击“保存”，完成修改。

* 删除菜单

选择需要删除的菜单，点击“删除”，点击“确认”，完成删除。



图4-24 删除菜单页面

##### 机构管理

用户以系统管理员身份登录系统，点击“权限管理”菜单中“机构管理”，打开机构管理页面；



图4-25 机构管理页面

页面中：“机构编号”是选填，为查询条件；

“机构名称”是选填，为查询条件；

“机构类型”是选填，为查询条件；

“上级机构”是选填，为查询条件；

点击“查询”，显示查询结果。

* 增加机构

点击“增加”，出现新增机构页面；



图4-26 新增机构页面

页面中：“机构编码”是必填项；

“机构名称”是必填项。

* 修改机构

选择一个需要修改的机构，点击“修改”，出现修改页面；



图4-27 修改机构页面

页面中：“机构编码”是必填项；

“机构名称”是必填项。

* 删除机构

选中需要删除的机构，点击“删除”，出现删除页面，点击“确认”，完成机构删除；



图4-28 删除机构页面

##### 用户管理

用户以系统管理员身份登录系统；点击“权限管理”菜单中“用户管理”，打开用户管理页面；



图4-29 用户管理页面

页面中：“登录名”指系统登录；

“姓名”指用户真实姓；

“行内雇员”指用户是否是行内雇员；

“是否禁用”指用户是否被禁用。

* 增加用户

点击“增加”，进入新增用户页面。



图4-30 增加用户页面

页面中：“登录名”是必填项，是登录系统的用户名；

“姓名”是必填项，是用户的真实姓名；

“密码”是必填项，是登录系统的密码；

“所属机构”是必填项，是用户所属机构；

点击“保存”后，完成新增用户。

* 修改用户

选择需要修改的用户，点击“修改”，进入修改用户页面。



图4-31 修改用户页面

页面中：“登录名”不能修改；

“姓名”可修改，是必填项；

“所属机构”可修改，是必填项；

点击“保存”后，完成修改用户。

* 删除用户

选择需要删除的用户，点击“删除”，点击“确认”，完成删除用户。



图4-32 删除用户页面

* 重置用户密码

选择需要重置密码的用户，点击“重置密码”，点击“确认”，完成密码重置功能。



图4-33 重置密码页面

##### 用户组管理

用户以系统管理员身份登录系统，点击“权限管理”菜单中“用户组管理”，打开用户组管理页面；



图4-34 用户组管理页面

* 增加用户组

点击“增加”，进入新增用户组页面；



图4-35 新增用户组页面

* 修改用户组

选择需要修改的用户组，点击“修改”，进入修改用户组页面。

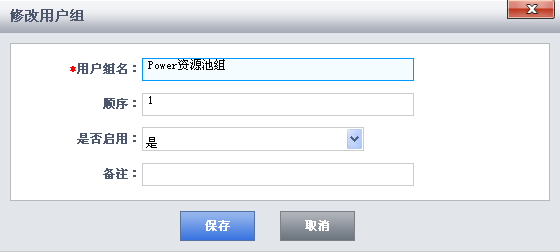


图4-36 修改用户组页面

* 删除用户组

选择需要删除的用户组，点击“删除”，点击“确认”，完成删除用户组。



图4-37 删除用户组页面

* 用户组报表授权

选择需要授权的用户组，点击“报表授权”，进入报表授权页面，选取报表树上的报表，点击“保存”，完成用户组报表授权。



图4-38 报表授权-用户组页面

* 用户组菜单授权

选择一条用户组信息，点击“菜单授权”，进入菜单授权页面，选取菜单树上的菜单功能，点击“保存”，完成用户组菜单授权。



图4-39 用户组菜单授权页面

* 用户组用户配置

选择一条用户组信息，点击“用户组配置”，进入用户组配置页面，选择左侧的机构树，右侧出现用户列表，选取用户信息，点击“保存”，完成用户组用户配置。



图4-40 用户组用户配置管理页面

#### 报表管理

##### 报表分类管理

用户以报表配置员身份登录系统，点击“报表管理”菜单中“报表分类管理”，打开报表分类管理页面；



图4-41 报表分类管理页面

* 增加报表分类

选中报表分类树，点击右键，系统弹出操作菜单，点击“增加”，系统进入新增报表分类页面；

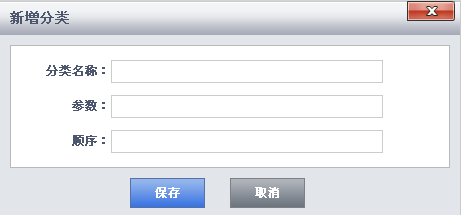


图4-42 新增报表分类页面

* 修改报表分类

选中报表分类树，点击右键，系统弹出操作菜单，点击“修改”，系统进入修改报表分类页面；

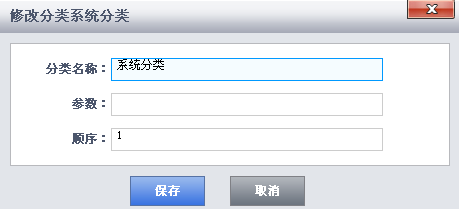


图4-43 修改报表分类页面

* 删除报表分类

选中报表分类树，点击右键，系统弹出操作菜单，点击“删除”，系统进入删除报表分类页面，系统弹出



图4-44 删除报表分类页面

点击“确认”按钮，系统检测分类下是否有报表存在，如果没有报表存在，系统提示，删除成功；否则，体系提示信息，分类下有报表，请先删除关联关系，



图4-45 删除报表分类页面

* 报表分类关系配置

选中左侧报表分类树，选择右侧报表列表，点击“保存”，完成分类报表关系配置；



图4-46 报表分类关系配置页面

##### 报表配置管理

用户以报表配置员身份登录系统；点击“报表管理”菜单中“报表配置管理”，打开报表配置管理页面；



图4-47 报表配置管理页面

* 新增报表配置

点击“增加”，进入新增报表配置页面

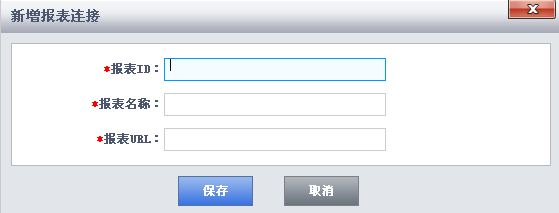


图4-48 新增报表配置页面

* 修改报表配置

选择需要修改的报表，点击“修改”，进入修改报表配置页面。



图4-49 修改报表配置页面

* 删除报表配置

选择需要删除的报表配置，点击“删除”，点击“确认”，完成报表配置删除



图4-50 删除报表信息页面

##### 报表参数管理

用户以报表配置员身份登录系统；点击“报表管理”菜单中“报表配置管理”，打开报表配置管理页面，选中一条报表配置信息，点击“参数管理”，进入报表参数管理页面；

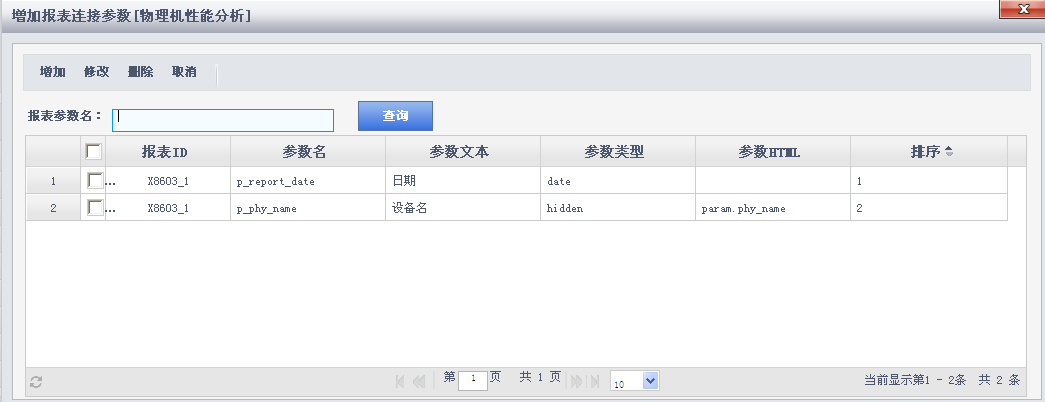


图4-51 报表参数管理页面

* 新增报表参数

点击“增加”，进入新增报表参数页面；



图4-52 新增报表参数页面

* 修改报表参数

选择需要修改的报表参数，点击“修改”，进入修改报表参数页面

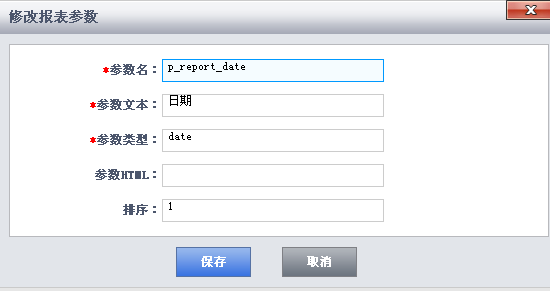


图4-53 修改报表参数页面

* 删除报表参数

选择需要删除的报表参数，点击“删除”，点击“确认”，完成。



图4-54 删除报表参数页面

#### 收藏夹管理

收藏夹管理包括个人收藏夹报表分类、新增收藏夹、修改收藏夹、删除收藏夹、收藏夹收藏报表信息功能；

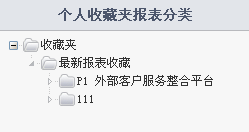


图4-55收藏夹报表分类页面

* 新增收藏夹分类

选择个人收藏夹报表分类树，点击右键，系统显示功能菜单，点击“增加”，进入新增收藏夹分类页面。

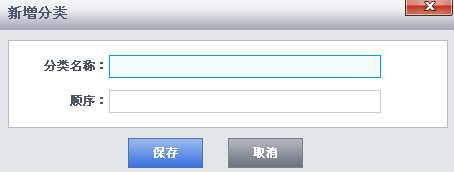


图4-56 新增收藏夹分类页面

* 修改收藏夹分类

选择个人收藏夹报表分类树，点击右键，系统显示功能菜单，点击“修改”，进入修改收藏夹分类页面。

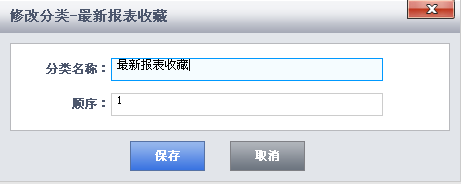


图4-57 修改收藏夹分类页面

* 删除收藏夹分类

选择个人收藏夹报表分类树，点击右键，系统显示功能菜单，点击“删除”，系统提示信息如下，



图4-58 删除收藏夹分类页面

点击“确认”，系统检测收藏夹分析下面是否有子节点，如果没有子节点，系统提示删除成功，否则，系统提示信息如下；



图4-59删除收藏夹分类页面

* 收藏夹报表关联信息

点击左侧收藏夹分类树，选择右侧报表树信息，点击“保存”完成收藏夹报表关联信息；



图4-60 收藏夹报表关联页面

### 系统的测试

测试是最常见的改善质量的活动，这种实践得到许多业界和学界研究，以及商业经验的支持。软件可以通过许多的方法进行测试，某些测试通常由开发人员进行，而有些则更多由专门的测试人员进行。

单元测试(Unit testing) 是将一个程序员或者一个开发团队所编写的，一个完整的类、子程序或者小程序，从完整的系统中隔离出来进行测试。

组件测试(Component testing) 是将一个类、包、小程序或者其他程序元素，从一个更加完整的系统中隔离出来进行测试，这些被测代码涉及到多个程序员或者多个团队。

集成测试(Integration testing) 是将两个或更多的类、包、组件或者子系统进行的联合测试，这些组件由多个程序员或者开发团队所创建。这种测试通常在有了两个可以进行测试的类的时候就应该尽快开始，并且一直持续到整个系统开发完成。

回归测试(Regression testing) 是指重复执行以前的测试用例，以便在原先通过了相同测试集合的软件中查找缺陷。

系统测试(System testing) 是在最终的配置下(包括同其他软硬件系统的集成)运行整个软件。以测试安全、性能、资源消耗、时序方面的问题，以及其他无法在低级集成上测试的问题。

测试本身并不能改善软件的质量。测试的结果是软件质量的一个指示器，但是结果本身并不改善软件质量。想通过更多的测试来改善软件的质量，就跟妄想通过天天称体重来减肥一个，你站在秤台上之前吃了多少东西，就决定了你称出来的重量是多少。同样的，你所使用的软件开发的方法决定了测试的时候会找到多少的错误。假如你希望减肥，不是应该买一个新的秤，而是应该改变你的饮食习惯，假如希望改善你的软件，仅用更多的测试是没有用的，你需要的是更高质量的开发[8]。

测试一般有两种方法：如果已经知道了产品应该具有的功能，可以通过测试来检验是否每个功能都能正常使用；如果知道产品内部工作过程，可以通过测试来检验产品的内部动作是否按照说明书的规定正常运行。前一个方法称为黑盒测试，后一个称为白盒测试[9]。

对于本系统，我采取第一个方法，即黑盒测试。做完一个模块，就进行测试，看所做的模块能不能实现所需要的功能。经过了先期的模块测试和后期的综合测试，本系统实现了规格说明书的所有功能，达到了预期的要求。

### 小结

本章详细介绍了系统主要功能模块的具体实现，包括数据模型设计、ETL设计和前端展现页面的主要相关界面的设计与说明，分析设计思路，执行步骤，并给出了每个功能的主要实现代码，还有为每个功能的最终执行结果配上截图，让我们看到更直观的效果。最后对完成的系统进行了测试，介绍了两种主流的测试方法，给出了测试结果，达到了预期的要求。

1. **总结与展望**

### 工作总结

通过设计和开发运行分析平台，我接触到许多软件开发方面的知识，包括HTML、CSS级联样式表、JavaScript、JQuery、Ajax、Spring MVC Struts2、MyBatis、Tomcat、Oracle、JSP和统一建模语言及建模工具等，这些技术扩展了我的知识，我在制作的过程中对它们的特点和应用方向等有了更深入的了解，并在实践中明白如何将它们融入应用，进而开发出界面美观且功能强大的应用软件。

本次系统设计和开发，我深刻的了解系统开发的整个过程，熟练掌握Web应用软件设计的思路及开发的各个环节。系统的需求分析，分析系统要达到什么功能，系统要划分为几个模块来设计，系统的具体功能应当如何去实现；分析完之后，考虑其涉及到的计算机专业知识，把问题细化，把大的问题划分为小的问题，然后逐个进行解决。

* 设计收获：

1.理解数据架构规划的分层设计思想；

2.掌握了ETL各个环节的处理机制和原理；

3.掌握了使用Spring3+Struts2+MyBatis的编程；

* 遇到的问题：

1. Struts2框架下的action中无法使用@Autowired自动注入Bean（运行时bean总是为null）

常见原因：

A、Spring的配置文件中组件扫描路径错误（即<context:component-scan base-package="xx.xx" />配置错误）。

B、@Autowired声明的beanName错误，bean的名称一定要在容器中注册过。

C、缺少struts2-spring-plugin-X.jar（X为版本号）。

2. Struts2 上传中文文件名文件下载后编程乱码，怎么解决？

解决办法是转码，经常用到的是下面的方式：

String filename = “中文文件名” ; //取得原中文文件名

filename = new String(filename.getBytes(“gb2312”),“ISO-8859-1”);//处理中文乱码

1. Struts2 怎样获取Ajax post 请求传递的数据？

从页面传对象到Action，Action用List接收，这样肯定是可以的，只要你的参数名字相同就行。如果真的不行，那就是用Struts2的类型转换功能，手动把JS数组串转换成Java的List，需要用Strutstypeconverter接口，从Action传对象到页面，把JSON串以流的形式写入页面，这样Ajax就会接到[10];

### 工作展望

由于个人能力有限以及实践经验欠缺，系统基本满足运行分析平台的基本功能需求，本系统不可避免有不足之处，如一些系统功能过于简单，系统的功能还算不上很完善。希望老师和同学们给予建议和指导。

1. **自我鉴定**

时光匆匆而过，转眼的时间我在北京理工大学继续教育学院计算机信息管理专业的学习生涯就快结束了，二年半的学习生涯和社会实践生活让我更充实了自己，为此，我特别珍惜在校园短暂的学习时光，也更提高自己自己的求知欲望，为以后的人生打下坚定的基础。

两年前，我就是因为自己的知识水平不够，不足以让我在当节奏飞快的社会中游刃有余，所以我选择继续深造，选择北京理工大学，选择计算机科学与技术专业，我孜孜不倦，不断地挑战自我，充实自我，超越自我，为实现自我的人生价值打下了坚实的基础。

两年多的成长与历练，我逐渐走向沉稳，养成了学习能力较强，求知欲强，爱好看书的习惯。两年多的大学生活塑造了一个健康，充满自信的我，自信来自实力，但同时也要认识到，眼下社会变革迅速，对人才的要求也越来越高，社会是在不断变化，发展的，要用发展的眼光看问题，自身还有很多的缺点和不足，要适应社会的发展，得不断提高思想认识，完善自己，改正缺点。要学会学习，学会创新，学会适应社会的发展要求。

在“德以明理，学以精工”八字校风的鞭策下，我努力学习，刻苦钻研、勇于进取、努力将自己培养成为具有较高综合素质的新时代毕业生。我尊敬老师，团结同学，以“热情、求实、尽职”的作风，积极完成学校和年级的各项任务，在实习期间，持着主动求学的学习态度，积极向指导老师学习，秉着“志不强者智不达”的信念，孜孜不倦地吸收知识为日后的学习、工作打下坚实的基础。不断追求，不断学习，不断创新，努力发展自我，完善自我，超越自我是我崇尚的理念。尽力做好每一件事是我个人的基本原则。

今天我为自己定了认证准则，虽然有些空大。但在我身经社会之前它确实是我心中所想，我愿意为它而努力。而且希望在我经历世事之后人能将这个准则作为心中的力量：诚实做人，忠实做事时我的人生准则，自强不息，勇于进取是我的在追求。

我坚信人必须保持奋进的心态。每天进步一小步，成就辉煌一大步。

**参考文献**

[1] 百度百科JQuery.

http://baike.baidu.com/view/1020297.htm。

[2] 百度百科 Struts2.

http://baike.baidu.com/view/1566725.htm。

[3] 百度百科 Spring MVC.

http://baike.baidu.com/view/1834982.htm。

[4](美) Deinum：《Pro Spring MVC With Web Flow》， Apress，2012年。

[5]陈雄华：《Spring 3.x 企业应用开发实战》， 电子工业出版社，2012年。

[6]王彦龙：《企业级数据仓库(EDW)原理、设计与实践》， 电子工业出版社，2006年。

[7](美)Ralph Kimball：《数据仓库工具箱(第3 版)》清华大学出版社，2015年。

[8] 杨少敏 樊双灵：《Struts2 Web开发学习实录》，清华大学出版社，2011年。

[9] 于涌 王磊：《精通软件性能测试与LoadRunner最佳实战》，人民邮电出版社，2013年。

[10](美)Steve McConnell：《代码大全(第2 版)》， 电子工业出版社，2009年。

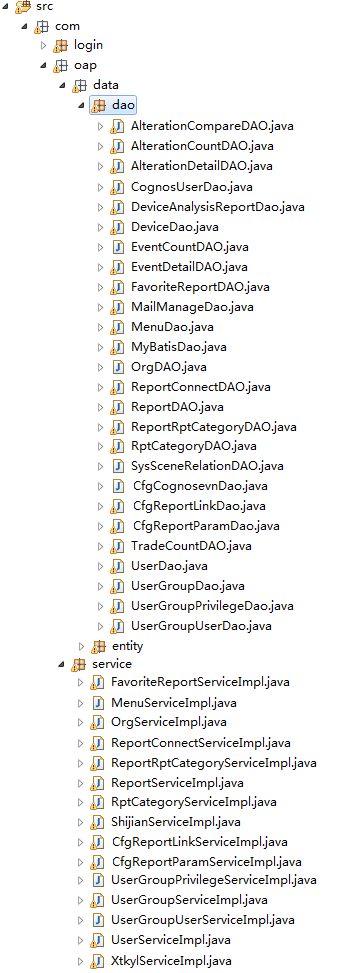
[11](美)Craig Larman：《UML和模式应用(原书第3版)》，[机械工业出版社](http://www.china-pub.com/search/power_search/power_search.jsp?key1=%BB%FA%D0%B5%B9%A4%D2%B5%B3%F6%B0%E6%C9%E7)，2012年。

[12] (美)：Ron Hardman Michael McLaughlin:《Oracle PL/SQL专家指南:高级PL/SQL解决方案的设计与开发》,清华大学出版社，2006年。

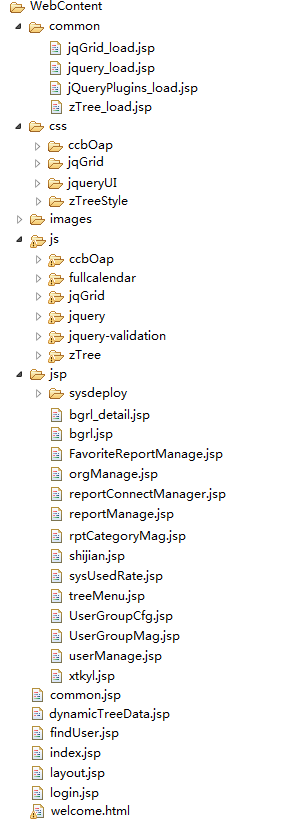
[13] (美)：Len Silverston:《数据模型资源手册》,机械工业出版社，2004年。

**附录**

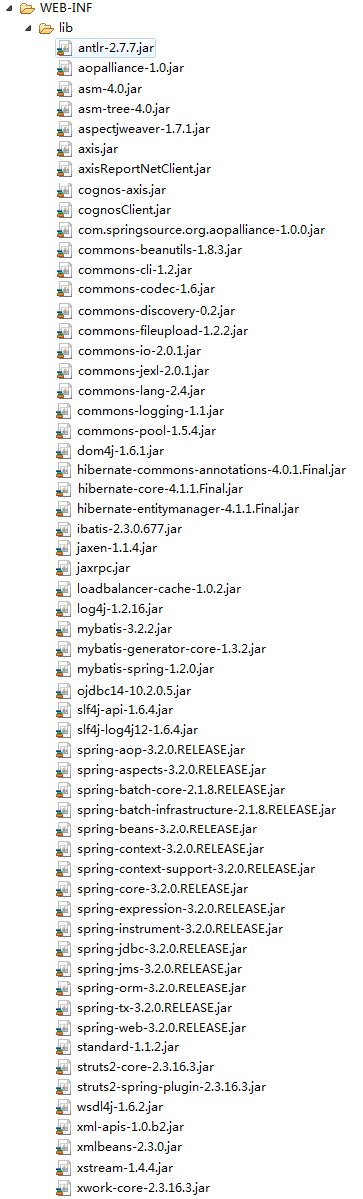
代码架构图：



图F1 Java类结构



图F2 WEB页面



图F3 引用包文件

**致谢**

本次毕业设计的最终完成离不开很多人的大力协助。真诚的感谢本次毕业设计的指导老师-姜增如老师，给了我自由选题的机会，对我的毕业论文最终完成给予了极大的支持和帮助。

感谢北京理工大学所有的教过我的老师，正是他们的无私奉献，才有我们今天的意气风发，我将以更加饱满的热情投入到以后的工作中去，充实自己，完善自己，向着自己的目标不断地前进，用我的所学去实现我心中的梦想!

最后，再一次感谢所有帮助过我的人们！