

南开大学

本科生毕业论文（设计）

中文题目：商务智能系统中多维数据可视化工具的设计与实现

外文题目：Design and Implementation of Multidimensional Data
Visualization Tool in Business Intelligence System

学 号：1212662

姓 名：朱晴宇

年 级：2012 级

专 业：软件工程

系 别：软件工程

学 院：软件学院

指导教师：谢茂强 副教授

完成日期：2016 年 5 月 31 日

关于南开大学本科毕业论文（设计）的声明

本人郑重声明：所呈交的学位论文，是本人在指导教师指导下，进行研究工作所取得的成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或没有公开发表的作品内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：

年 月 日

本人声明：该学位论文是本人指导学生完成的研究成果，已经审阅过论文的全部内容，并能够保证题目、关键词、摘要部分中英文内容的一致性和准确性。

学位论文指导教师签名：

年 月 日

摘 要

在大数据时代,多维数据的可视化成为了解和分析大量数据信息的重要支撑方法,它能够为人们营造直观的、交互的可视化环境,让分析者在有限时间内高效地处理结构复杂的海量数据集,为决策者提供可靠依据。

本文通过对系统用户需求的分析,从不同用户的角度设计与开发了服务于商务智能系统的多维数据可视化工具。为了提供友好的用户体验,设计的工具界面美观、操作简单。在全面分析了可行性之后,用 Java 语言与 MVC 框架开发此工具,实现了多维数据集到个性化图形表达的转变,让抽象的信息具体化、直观化。该工具的特点是用户可以自由选择商业分析角度,从而得到个性化、具有交互功能的多维数据可视化图形,简单易懂的表达形式能够很好地帮助用户进行精准的商业决策分析。

关键词：商务智能；多维数据可视化；Java；MVC 框架

Abstract

In the era of big data, multi-dimensional data visualization becomes an important method to understand and analyze large amounts of data information, it can create an intuitive, interactive visualization environment and help people with efficiently handling the complex structure of large data sets in a limited time. Moreover, it provides a reliable basis for decision makers.

Based on the analysis of the user requirement for the system, this paper designs and develops a multi-dimensional data visualization tool from different angles for different users. In order to provide a user-friendly experience, this tool owns a beautiful interface and a simple operation. After a comprehensive analysis of the feasibility, Java language and MVC framework are used to develop the tool, it achieves the transformation between multi-dimensional data and personalized graphs, so that the abstract information becomes to be concrete and intuitive. It can help users with carrying on business decision analysis. Feature of this tool is that users can freely choose business analysis points from different choices, and then users would obtain personalized and interactive visualizations, It can be a good helper when users make business decisions.

Keywords: Business intelligence; Multi-dimensional data visualization; Java; MVC framework

目 录

第一章 引言	1
第一节 研究背景	1
第二节 国内外成果综述	3
第三节 论文的主要研究	4
第四节 论文的内容安排	5
第二章 Data graph 需求分析	6
第一节 工具简介	6
第二节 工具需求	6
第三节 可行性分析	8
第四节 功能性需求	9
第五节 工具非功能需求	11
第三章 Data graph 整体设计	12
第一节 模块划分设计	12
第二节 开发架构设计	14
第三节 多维数据模型设计	16
第四节 工具开发环境设计	18
第四章 数据封装与调用	21
第一节 数据请求响应	21
第二节 数据查找	22
第三节 数据封装与转换	23
第四节 数据读取与映射	25

第五章 Data graph 工具实现	27
第一节 交互选择功能实现	27
第二节 传统图形实现	29
第三节 热点图与地图实现	34
第四节 系统其他工具展示	37
第六章 总结与展望	41
第一节 总结	41
第二节 展望	41
参考文献	42
致谢	44

第一章 引言

在现代的商务智能系统中，数据可视化是了解和分析系统中大量信息的重要支撑方法。商务智能系统通常将市场销售、商场部门和顾客消费等各类数据集合在一起，当收集到的原始数据量达到一定规模之后，随之会出现数据分散、杂乱的现象，这一特点给商务分析带来了巨大的挑战。正是数据可视化这一技术，通过视觉冲击和人机交互的方式给信息接受者和分析者提供了一个高效的研究平台。然而，如何精准、有效地展示复杂数据之间的内在关系是一门结合了艺术设计和科学技术的学问。当完成庞大原始数据的收集工作之后，商务智能系统中的数据可视化需要包含用户交互性强、数据显示高维性以及展示方式多样化等特点，为系统的用户提供快速精准的商务决策。

第一节 研究背景

商务智能(BI, Business Intelligence)系统是一个数据驱动的决策支持系统，它将数据收据，数据存储和知识管理结合在一起，并且通过分析为决策过程提供输入信息^[1]。商务智能这个名词起源于 1989 年，在此之前，它的很多特征被包含在执行信息系统中。以计算机为基础的商务智能采用大型数据库，通常将数据存放在数据仓库或者数据集中，为系统提供信息来源以及为之后复杂的商务分析提供可靠基础。系统的分析功能十分广泛，从提供简单的报表，数据切割、钻取，到实时信息查询，在线分析以及预测。近些年商务智能系统不断发展，还包含了企业绩效评价，商业活动监控等功能。除此之外，商务智能系统的用户由企业员工扩展到其他组织机构成员。商务智能技术能够在商业过程成为越来越关键的一部分^[2]。

可视化技术将科学计算的结果以图形的方式表达出来^[3]。它把人们捕捉到的复杂信息转变为人们通俗易懂的形式^[4]。从最广泛的角度上定义，可视化技术范围包含任何创建图片以及展示抽象数据的技术。它被人们广泛应用于商业、医疗、航空航天和新闻报道等行业，为人类社会的发展和经济效益的增长提供了革命性帮助。

南丁格尔玫瑰图可以称得上是最具有影响力的数据可视化图形^[5]。在战争期间伤亡人数十分惨重，英国护士和统计学家南丁格尔将这些数据收集整理，用统计学图表方式创作出一幅色彩缤纷的玫瑰图，亦称鸡冠花图，见图 1.1。这张玫瑰图用来揭露军医院季节性的死亡率，它表示在 18000 位伤亡战士中，有 16000 位是由于在后方医院中感染疾病致死而非因战争枪伤。南丁格尔这张具有开创型的图表促使英国政府开始重视战地医院的卫生状况以及医疗条件，由此挽救了上万名战士的生命^[6]。

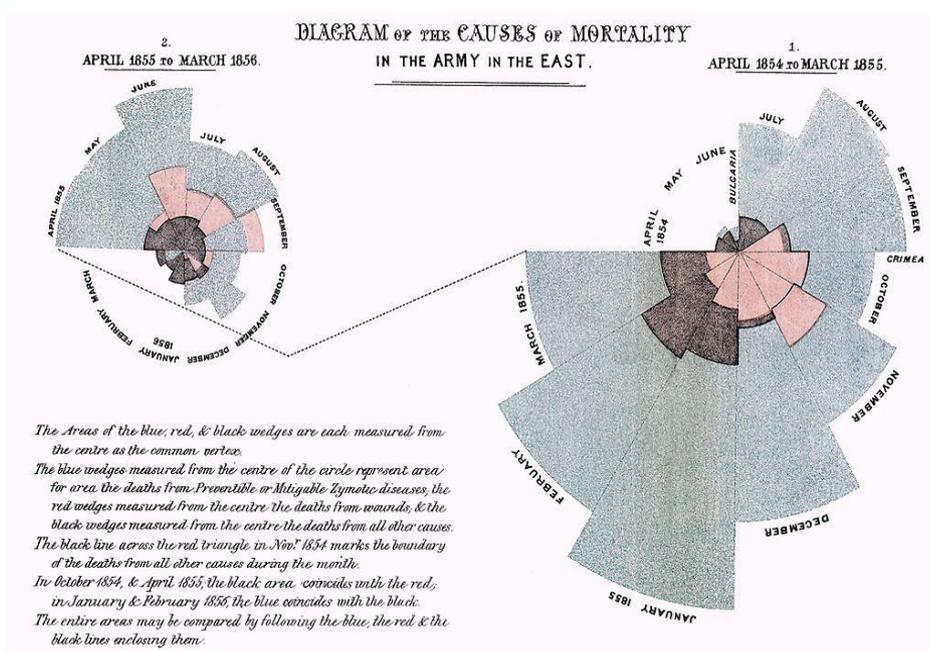


图 1.1 南丁格尔玫瑰图，来源维基百科。

数据可视化是信息可视化的一部分，它包括任何一种将数据转化为可视化信息的技术，旨在帮助人们对数据有更充分的理解，它是一个可重复的过程^[7]。该技术利用计算机图形学技术将复杂的、多维度的数据集合进行数值分析并且从中提取有价值的信息^[8]。传统意义上的数据可视化过程是将原始数据过滤，选择可利用、感兴趣的部分，将其映射到图形化的形式上，然后生成图片、动画或者其他可视化产品。研究人员通常在高性能的计算机上模拟生成大量的数据集，然后利用数据可视化技术捕捉数据特点和趋势。

近些年，数据可视化技术被开发者不断研究并将开发的各类炫目的数据可视化工具发布在网络中以便于人们再次研究与开发。将多维数据可视化技术嵌入到商务智能系统中，为决策者提供独特、便捷的分析方法，对于开发者来说是一项

充满意义的工作^[9]。

第二节 国内外成果综述

随着多维数据可视化技术快速发展,国内外许多软件公司将这一新鲜血液注入商务智能平台中,开发了一系列具有强大数据可视化功能的商务智能系统^[10]。例如: SAP Business Object 和 Smartbi。

SAP Business Object 是由全球领先的 BI 厂商 SAP 开发,具有完整和开放的解决方案。它由商务智能、数据仓库以及企业项目管理等套件组合为一个整体,并且兼容 Oracle、IBM 和 Microsoft 等其他技术。它可以即使生成的可视化报表并以各种格式传递给任意用户,例如 Office 文档、文本和 PDF 等。

Smartbi 是由我国思迈特软件有限公司开发的软件。它与金融、制造业与政府各单位合作为各类用户提出灵活的商业解决方案。并且可以使用拖拽的方式把数据挖掘过程用视觉方式展现出来。用户可以任意选择参数、图形等功能,能够在各类移动系统设备上显示图表。

根据统计结果表明, Tableau 软件是商务智能的数据可视化市场的领先者。该软件任务处理流畅、快速并简单,在所有数据可视化中显示平滑的工具提示不会因为数据的复杂程度受到影响。它可以支持移动设备与网站平台的同时登录状态,通过搜索功能查看浏览次数最多的视图。新增的功能包括以地图的形式展现 2016 年美国人口统计,通过图表预测异常季节,利用选项卡来添加不同的颜色以便于用户区分等。

日前功能强大并且受众多开发者喜爱的多维数据可视化 JavaScript 工具库有如下几种: arbor.js、Paper.js 和 D3.js 等。它们针对不同的图形化创造需求,功能各具独特性。

Arbor 是一个面向网站开发者,建立在 jQuery 之上的图形可视化库。Arbor 与那些以满足各种需求为目的的框架不同,它提供了一种高效的、力导向的布局算法。Arbor 为用户提供了非常真实的屏幕绘图体验,可以将 canvas 和定位的 HTML 元素等与其结合。开发者不需要把大量的时间花繁琐的布局上,只需要关注事物的本身就可以创造出独一无二的图像数据和视觉风格,它所提供的各种展示方法都可以满足项目的表现需求。其官方地址为: <http://arborjs.org/>。

Paper.js 是一个开源的矢量图形脚本框架，它运行在 HTML5 Canvas 之上，为用户提供一个清晰的场景图，并具有强大且丰富的向量图形和贝兹曲线创作功能。它拥有设计良好和连贯性强的编程接口，使初学者很容易就能掌握。Paper.js 的网站是：<http://paperjs.org/>。

D3.js 是基于数据操作文档的 JavaScript 库。它能够帮组开发者通过使用 CSS 和 SVG 等，从而让数据富有鲜活的生命力。D3 能够帮助开发者将零散的数据绑定到一个文档对象模型中，然后把数据驱动转换应用在文档中，它将强大的可视化元素与数据的驱动方法结合，来进行文档对象模型操作^[11]。D3 能够提供解决问题的关键技术，即它能够高效地操作以数据为基础的文档。因此能够避免一些局限性，而提供很强的灵活性。D3 的处理速度及其快，它可以支持大型数据库并且可以展示动态的交互效果与动画图像。官网：<https://d3js.org/>。

综上所述的商务智能平台主要面向于专业的商务分析技术人才，需要对数据仓库和多维数据的知识有基本的了解。但是我们希望能够开发一款不仅能够使有背景知识的人员操作还能够为其他人群，如，普通消费者提供该企业的各类信息，这时候就需要系统提供操作方法简单的数据可视化。除此之外，大部分的商务分析平台能够提供给用户使用的可视化图形主要是传统的柱状、折线和饼状三类，在这基础上我们希望能够为用户提供更多个性化的图形，例如：热点图和地图等。

第三节 论文的主要研究

在这篇论文中我们主要研究如何基于数据可视化的 JavaScript 框架来设计与实现能够分析和展示多维数据的可视化工具，此主要着眼于商务智能系统。我们采用新概念的 Foodmart 超市为研究实例，Foodmart 数据库记录着该跨国食品超市每天的销售、顾客、员工和库存信息。我们围绕 Foodmart 中多维数据集信息，建立名称为 Flash analysis 的商务智能系统，其中的多维数据可视化工具称为 Data graph。该工具不仅面向具有专业背景知识的企业的管理者、员工，还为该企业的普通消费者提供感兴趣的可视化产品。由于我们开发的 Flash analysis 系统面向的用户来自各行各业以及各年龄段，这其中很多人对移动端的 app 操作不熟悉，所以该系统是一个基于 web 开发且可适用于各类浏览器的商务智能数据分析网站。它把联机分析处理技术、多维数据可视化技术以及查找算法三维于一体，

系统地分析企业中庞大的信息。该 Flash analysis 系统具有很强的灵活性，可以支持不同的操作系统，比如 Windows 8 和 Mac OS X 10。当用户输入账号信息登录进入该网站，自由选择想要的分析方式，就可以根据不同的需求，随后便可得到便于商业预测和规划等的可视化产品。我们开发的 Data graph 工具基于 D3.js 框架，它兼容性强大并且能够很好地服务于 web 端，能够提供大量个性化图形，并且扩展性高。

Data graph 工具的实现主要研究内容分为以下几点：

1. 如何处理、分析收集到的零散原始数据？
2. 对整理好的数据集，我们如何利用 D3.js 框架将数据的不同维度信息映射到可视化图形中？
3. 在进行多维数据可视化方法设计时，我们应该选择哪种图形简洁、准确地展示数据？
4. 不同的图形显示，我们需要掌握哪些技术，灵活地展示不同类型的数据？
5. 我们如何展现多维数据的维度和粒度特点，从不同角度捕捉数据之美？
6. 数据量增大时，应采取何种方式存储、读取数据提高系统的处理效率？
7. 如何为用户提供友好的界面和良好的交互体验？

第四节 论文的内容安排

这篇论文以多维数据可视化技术为主要研究内容，将其运用在商务智能系统中以提供多种多样的数据可视化服务，从该商务智能系统的多维数据可视化技术的需求分析、框架设计、多维数据处理和系统的实现等开发角度进行详细展开。

论文的内容安排包括 6 个章节。第一章的引言部分围绕 w 论文的背景知识和研究内容展开。第二章着眼于多维数据可视化的需求分析，并介绍功能性需求与非功能性需求以及系统的可行性。第三章有关系统中 Data graph 工具的设计思路，包括模块划分、流程、架构以及多维数据模型的设计。第四章是详述多维数据的封装与过程，解决如何响应前端界面数据请求、查找数据、封装调用数据、解析与映射数据问题。第五章是技术的实现和界面部分，展示 Data graph 工具中各个图形的输出结果和系统的整体界面。第六章是总结全文以及对本文未来的展望。

第二章 Data graph 需求分析

第一节 工具简介

我们为商务智能平台开发的 Data graph 多维数据可视化应用工具，是一个辅助普通顾客和商务专业决策者高效分析 Foodmart 超市的地域分布、产品销售、职员信息的工具，它可以提取该企业中有价值的信息。

对于没有专业背景知识的普通用户，可以根据前端界面的提示，根据自身的需求选择不同的维度，粒度和指标。从而获得具有个性化的热力图、柱状图、折线图、饼图和地图。这些图形不仅可以静态的展现数据信息，还能够提供交互式动态的信息提示，能够满足不同类型用户的需求。

我们用 Data graph 工具将该企业最具代表性的图形产品作为事例发布到平台的首页中，当用户进入产品的详细页面，可以获得该图形的分析结果，并为用户提供操作提示。

第二节 工具需求

2.2.1 Data graph 的用户需求

我们在 Flash analysis 系统中采用 Foodmart 跨国超市已有的每日记录来填充的我们数据库，在此基础上我们实现了 Foodmart 超市的数据仓库、多维数据可视化与信息查询三大模块功能。因此该 Data graph 工具的用户主要由系统管理员和普通用户组成。

该系统管理员主要负责系统的日常维护与管理。系统管理员根据用户注册时填写的信息进行身份验证，新注册的用户身份验证通过之后，系统管理员需要针对不同角色的用户，设置不同的权限。由于该系统的普通用户又包括企业员工、顾客和企业合作，为了保护好企业的数据库资产，系统管理员要在用户的商业核心数据上增加访问权限，只有系统管理员指定的用户才能自由访问。系统管理员还可以设定系统运行时需要的相关数据库，配置好系统分析的数据源。对数据库中

的异常数据进行删除和恢复，把新记录添加到原数据库中。系统管理员还能管理用户的信息，当企业员工有人事变动时，管理员可以重置该用户角色，除此之外，系统管理员可以对账号进行增删。

Flash analysis 系统为企业决策者提供了时间、地区、销售和人事四大类分析模块，他们在该系统中需要完成的工作仅是选择感兴趣的数据可视化表达图形，进入上述所说的四类分析模块中的其中一类，再动态确定多维数据的维度，粒度与指标，随后得到的可视化产品将辅助他们进行商业分析，除了能够获得传统的柱、条和饼状图外，用户还希望选择更多个性化地多维数据可视化图形，例如热力图和地图，并且还能在不同的图形之间进行转换，多种视觉角度挖掘数据的商业价值。

不具有核心数据访问权限的普通用户，在该系统中可以查询企业的公开数据，查看公开数据的可视化图形和企业的介绍文档。这些功能可以让合作者对该企业有一个整体上的了解，针对企业对于那些不太能理解传统统计报表的普通用户，也可以从这些个性化、简单易懂的图形中快速地掌握该企业的整体商业信息。

2.2.2 用例图

1) 根据上一小节的内容分析，该系统管理员用例图如图 2.1 所示：

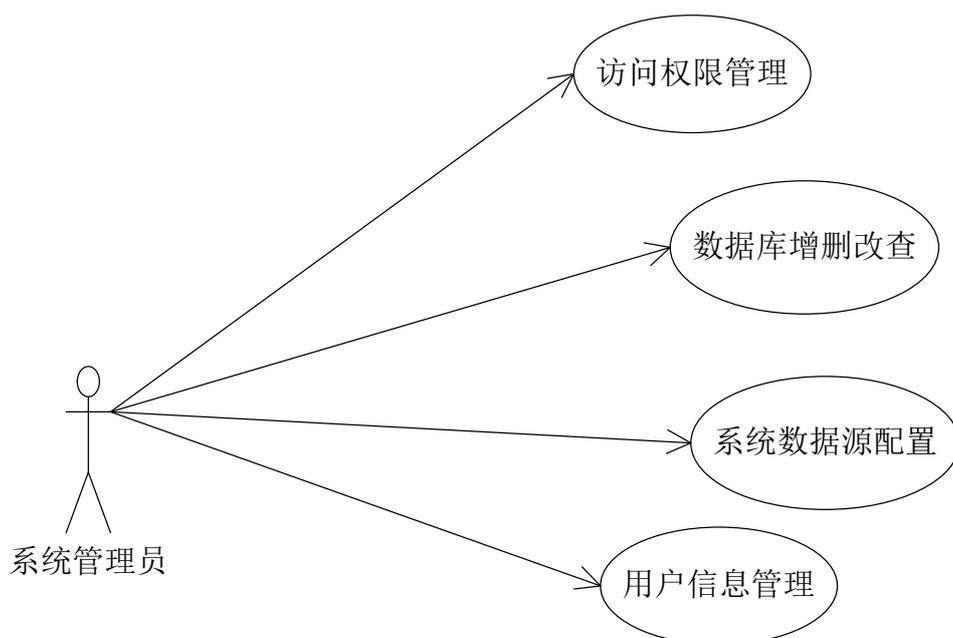


图 2.1 系统管理员用例图

2) 可以访问企业核心数据的商业决策者用例图如图 2.2 所示

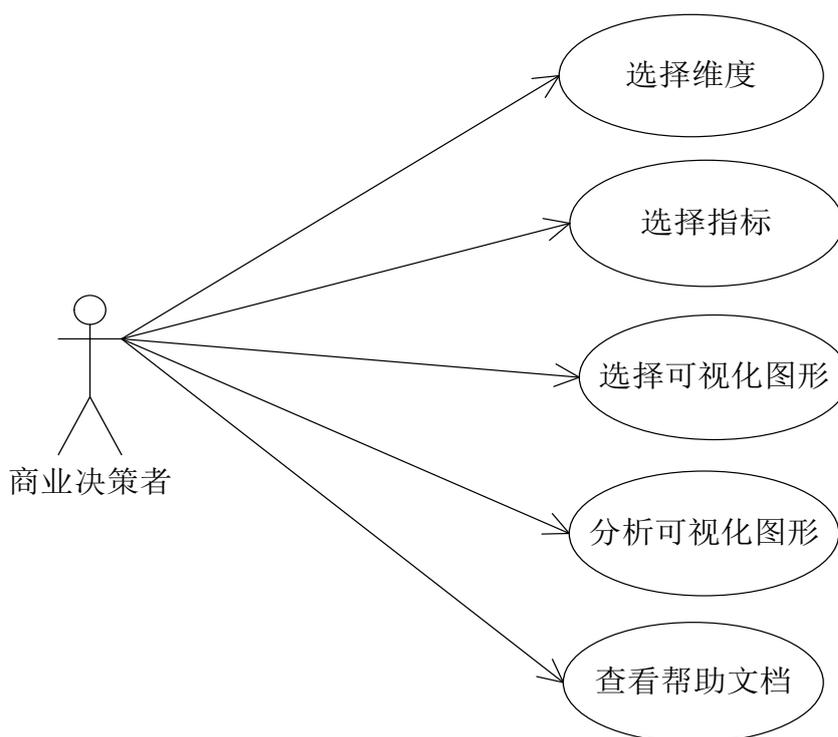


图 2.2 商业决策者用例图

3) 没有权限访问企业关键数据的其他用户用例图，如图 2.3 所示

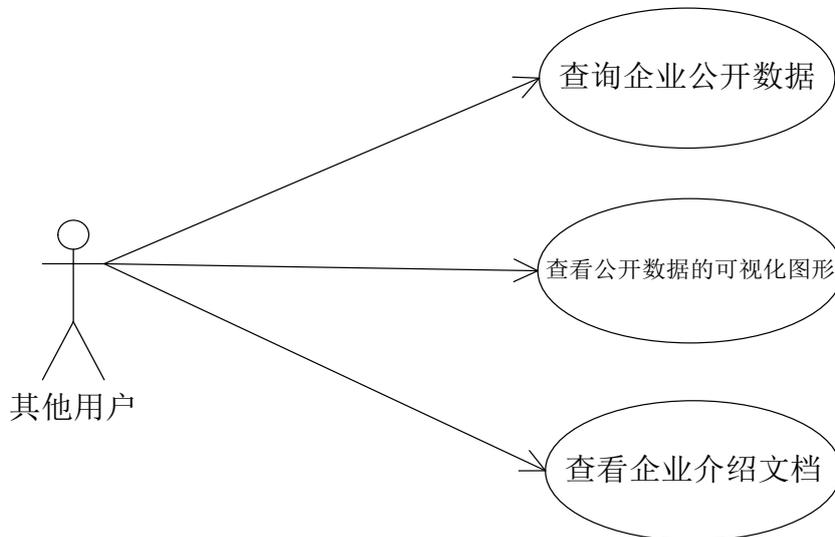


图 2.3 其他用户用例图

第三节 可行性分析

在这篇论文中所研究的 Data graph 工具是我们开发的 Flash analysis 商务智能

系统的一部分，它将作为该系统的子系统与数据仓库分析、信息模糊查询两个子系统组成一个完成的商务数据分析平台。Data graph 的优点在于它能够面向专业分析人员和非专业的普通客户，能够动态获取该子系统中用户的指令，并高效地封装后台数据，显示出个性化的数据可视化结果，这一小节中，我们立足于整个 Flash analysis 平台，通过以下两个角度分析 Data graph 工具的可行性。

2.3.1 Data graph 工具技术可行性

在整个项目中，我们是基于 Web 实现的 Flash analysis 商务智能系统，该系统通过 Java 语言搭建 MVC 框架来实现后台数据库与前端界面的连接。这一技术的实现在如今已经非常成熟，因此不会成为整个项目的难点。其次，针对该系统中的 Data graph 工具而言，我们是利用时下十分流行且扩展性极高的 D3 可视化框架，这一技术在近些年被开发者大量利用，代码编写简单。编写 Java 程序生成动态 JSON 文件也是当下使用广泛，性能稳定的技术。因此，在 Flash analysis 系统实现多维数据可视化工具过程中遇到技术难点都可获得解决方案。

2.3.2 Data graph 工具操作可行性

该程序面向的用户主要是企业的专业决策分析人员和普通的消费者，他们绝大部分不会编程或系统开发。而我们开发的 Flash analysis 系统基于 Web 网页，以图标点击和简单的输入为主。用户只需要简单的注册、登录就可以进入平台，菜单栏配有清晰的图标和文字提示，用户可以轻松上手。

第四节 功能性需求

2.4.1 维度的选择功能

Flash analysis 系统主要提供四类分析模块分别是时间、地区、商品销售以及企业人事。除此之外该系统的 Data graph 工具为用户提供了四类数据可视化图形，它们是，柱状、折线、饼状、热点图以及地图。在用户获得可视化图形结果之前，需要在界面进行维度的选择。

首先用户通过点击需要选择的某类图形的图片，系统链接到对应图形的生成界面，一组下拉框为用户提供四个分析模块的选择，用户在下拉框选择某个模块的之后，会联动出现一组有关维度、粒度的下拉框，确定好维度和粒度之后，再点击相关的按钮，随后可以展示图形结果。

例如用户点击柱状图形图片，进入图形生成页面，从四个分析模块的单选框中选择了时间分析，这时候，会动态出现时间、商品两个维度的对应粒度下拉框。例如，时间可细分为每天的日期、年、月、季度等；商品维度中分为冷冻食品、早餐类食品和日用品等维度。再点击指标按钮，例如总销量。

这些交互功能是为了给用户提供了多维数据的分析步骤，灵活地展现个性化图形。

2.4.2 图形的动态展现功能

该工具需要根据用户在前端交互选择的结果，及时响应，将其传到后台程序中，根据要求从数据库中动态地提取相应的数据，获得的图片需要图文结合，准确地显示用户选择的维度、指标等信息。以使用户读懂图片所表达的信息。

例如，按照上一小节中用户通过交互选择得到了柱状图，在获得的柱状图中，图形的颜色要能够给用户提供良好的视觉效果，如用橙色来填充柱形。当用户鼠标停留在具体的柱形条上时，会有浮动提示框交互显示该柱形条的信息。柱形图的横纵坐标分别代表两个不同的维度，例如销量和时间。

2.4.3 图形转换功能

该工具需要提供在不同类型的图形中灵活切换的功能，例如：折线图变换为柱形图，热力图变为折线图，柱形图换成饼图等。为用户提供多种选择。

2.4.4 用户注册、登录功能

整个 Flash analysis 系统需要新用户进行简单的注册，再登录之后，方可以进入系统，再能使用系统各模块中的功能。以便确保企业信息安全。

第五节 工具非功能需求

2.5.1 Data graph 工具接口

从系统的开发人员角度分析，需要有配置好 Eclipse、Tomcat、SQL Server 以及 MySQL 开发环境的计算机。

从该系统的用户角度分析，由于该系统的 Data graph 工具是基于 D3 技术开发的，该技术使用了 SVG，它无在 IE8 和更低版本的 IE 浏览器中产生图形效果，因此用户需要安装有 IE9 以上以及其他浏览器，通过网址访问链接到 Flash analysis 系统的登录界面^[12]。

2.5.2 Data graph 工具稳定性

我们现在开发的 Flash analysis 分析平台是靠 Foodmart 数据库获得网站的数据支撑，Foodmart 数据库里现有的数据是 1997 和 1998 两年的数据，数据的总量不是很庞大，考虑到数据会不断增添，为了使 Data graph 工具能够有良好的处理能力，我们将读取的 JSON 数组存放在 JSON 文件中，即将数据的存放位置从内存转移到硬盘中，以便减少内存的消耗，保证平台的稳定性^[13]。

2.5.3 Data graph 工具扩展性

我们开发的 Data graph 工具后台采用 MVC 的框架，通过 Servlet 接口把后台数据传输到前端的 jsp 页面，当后台数据库更新时不需要改变代码，便可自动获得更新的数据。我们前端 jsp 页面采用的 D3 框架也给开发者提供了良好的接口，具有很高的代码复用性和扩展性。

2.5.4 Data graph 工具可维护

从开发人员角度分析，Data graph 工具还需要具备易测试性。易测试性包括系统的各属性可以动态改变，前端的界面生成，提高工具的可维护性可以节约开发人员所投入的时间成本和精力。

第三章 Data graph 整体设计

第一节 模块划分设计

在这一章节中，我们从 Data graph 工具的模型设计、开发架构设计、多维数据模型设计和工具开发环境设计四个小节详细介绍 Data graph 工具的总体设计。第一小节详述该工具的模块划分与其工作流程。

3.1.1 模块划分设计图

下图 3.1 具体展示了工具的模块划分结构。该工具分为两个主要模块：传统统计学图形模块和多维数据个性化图形模块。用于数据信息的可视化，为数据提供图解分析方法。在提供良好支持。

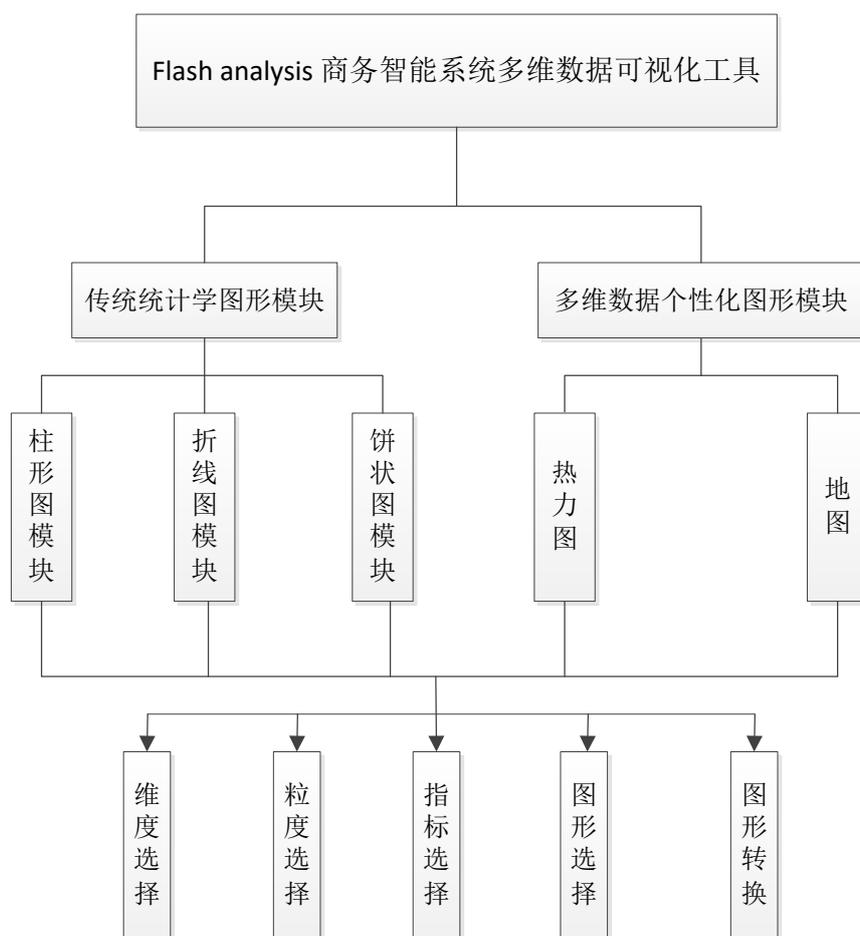


图 3.1 Data graph 的模块功能体系图

3.1.2 模块工作流程

我们把 Data graph 工具按照三类不同用户对图形种类的需求，分为两大主要模块，分别是传统统计学图形模块和多维数据个性化图形模块。在传统统计学模块中有柱形图、折线图和饼状图，在多维数据个性化图形模块则包括热力图和地图。

用户在系统首页分别选取不同的图形预览图，然后链接到不同的图形生成界面中。所有的图形生成页面，都具有以下功能：多数据的维度选择功能，粒度选择功能，指标选择功能，图形选择功能和图形间相互转换功能。整个工具的详细工作流程设计如下：

用户登录系统后在首页可以看到四类图形，以图片预览的方式展现出来。点击对应的图形的预览，用户可以进入多维数据定义界面。我们提供的图形不仅有传统的二维图形，例如柱、饼和折线图，还有个性化的可视化图形，分别是地图以及热点图。

在图形生成界面，最上方分别是时间、地理、商品与人事四类维度的联动下拉框。例如选择时间维度后，紧挨着的右边会出现关于时间的年、季度、月份、星期、每月第几号、每年第几月、每年第几周 7 个粒度。四个下拉框至少需要选择两个才能够显示出数据可视化的图形结果，再选择销量或者销售总额等指标，即可生成图形。

当用户完成交互的下拉框选择之后，jsp 页面接受来自用户的数据请求，按照指定的结构把数据到服务器端。服务器开始执行数据库的访问和查询命令，把获取到的数据信息返回到 jsp 页面，再通过 D3 读取、解释数据，把数据属性映射到图形的属性上。界面可以向用户动态展示图形信息。在图形的生成界面，用户还可以查看到图形的简单介绍信息，比如颜色的深浅代表的含义。其中有些图形可以给用户显示动态的信息，便于用户对可视化产品的阅读。

当用户想进行图形转换时，例如用户获得 1997 每天的饮料类商品销售量柱状图后，希望得到相应的折线图进行对比，这时候用户在图形转换中的一组单选框中点击折线图按钮，再按确定，数据的传送过程和下拉框的一样，随后页面自动刷新，用户可以访问到转换后的折线图。

第二节 开发架构设计

3.2.1 开发架构设计图

Data graph 工具采用 B/S 架构开发，分为前段浏览器端、服务器端和数据库端三个层次，其开发架构设计图如图 3.2 所示：



图 3.2 Data graph 的开发架构设计图

3.2.2 前端

Data graph 工具的前端界面是工具最主要的部分，它提供了全部的交互选择、响应需求，数据分析与映射以及图形动态生成等功能。支持 D3 技术的浏览器都能够展示丰富的图形效果，并具有很高的可靠性与稳定性。我们在前端采用 Java 开发中常用的 MVC 框架，可以将页面展示、业务逻辑以及底层数据分离，读取

传输数据的 JSON 文档。

3.2.3 服务器端

前端页面获得用户的请求之后，会将指令传到工具的服务器端，这时候，服务器端的代码会进行数据查询与数据可视化产品的调回。前端的 jsp 界面动态响应可以请求，将对应的信息传到查询语句中，当请求比较多时，我们设计在服务器中实现数据的解析、存储，例如，我们在 Servlet 层生产 JSON 文件而不是 JSON 数组，目的是为了减少内存和前端的负载，保证系统的效率。我们在 dao 层完成对数据库的数据查找工作，用 Servlet 层实现与界面端 JSP 的数据交互。

3.2.4 后台数据库

Flash analysis 系统中采用了两个数据库，MySQL 数据库是用来存储用户信息表，存储用户 id，头像，昵称、密码和邮箱等字段，用户 id 是自增长属性，用户角色分为系统管理员和普通用户，用户的实体图如下图 3.3 所示：

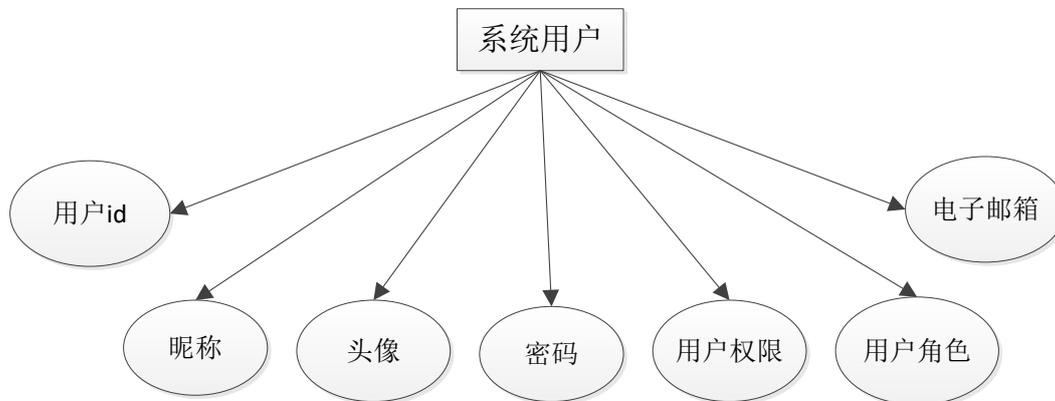


图 3.3 用户实体图

具体设计如下表 3.1 所示：

表 3.1 用户表

属性名	类型	长度	空否	默认值	主/外键	注释
Userid	INTEGER	10	否	Not null	P	用户 id
Userpho	VARCHA	20	否	Not null		头像
Userstatus	INTEGER	2	否	Not null		用户权限
nickname	VARCHA	20	否	Not null		昵称
Password	VARCHA	10	否	Not null		密码
email	VARCHA	20	否	Not null		电子邮箱

role	VARCHA	20	否	Not null	用户角色
------	--------	----	---	----------	------

SQL Server 负责存储 Foodmart 数据库。在服务器端的 po 层，我们把数据存储为序列化的对象，完成对数据库中不同表、不同属性的映射。

第三节 多维数据模型设计

在这一小节中，我们主要从时间、地区、商品类别和部门四个维度，用 Foodmart 超市 1997 年和 1998 年的销售为事实，设计多维数据模型。该多维数据模型是雪花型，商品类别维表通过商品维表连接到销售事实表上，见图 3.4：

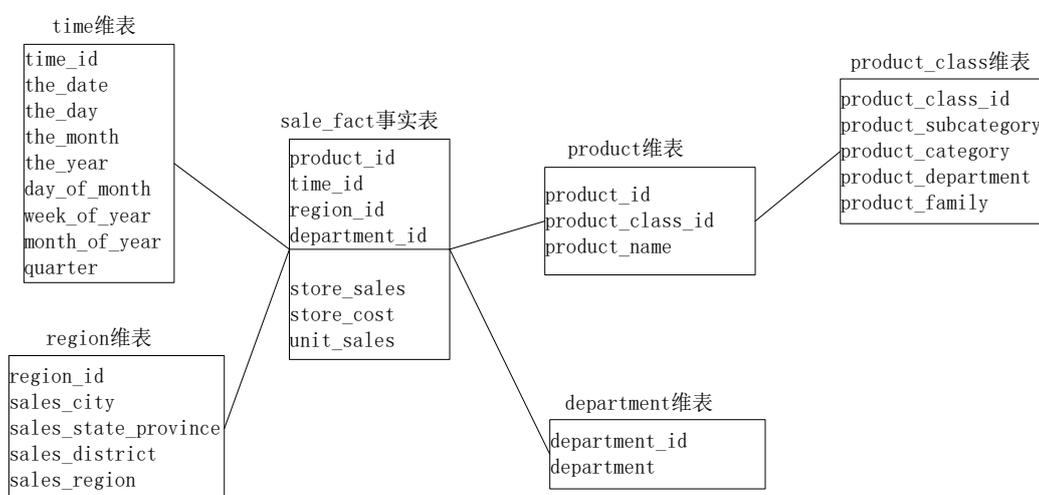


图 3.4 多维数据雪花型模式图

(1) 在 Foodmart 数据库中的时间维度表 (time_by_day 表，见表 3.2) 中，我们把时间作为维度，包含有年、季度、月、周、日、每月第几号、每年第几周和每年第几月共 8 个粒度。

表 3.2 时间维度表

属性名	类型	长度	能否 Null 值	主/外键	注释
time_id	int	默认	否	P	时间 id
the_date	smalldatetime	默认	是		年月日
the_day	nvarchar	15	是		周
the_month	nvarchar	15	是		月
the_year	Smallint	默认	是		年
day_of_month	smallint	默认	是		每月第几号
week_of_year	float	默认	是		每年第几周
month_of_year	smallint	默认	是		每年第几月
quarter	nvarchar	默认	是		季度

(2) 在地区维度表 (region 表, 见表 3.3) 中, 提取销售城市、州省、行政区、地区和国家五个粒度。

表 3.3 地区维度表

属性名	类型	长度	能否 Null 值	主/外键	注释
region_id	Int	默认	否	P	地区 id
sales_city	nvarchar	50	是		销售城市
sales_state_province	nvarchar	50	是		销售州省
sales_district	Nvarchar	50	是		销售行政区
sales_region	nvarchar	50	是		销售地区
sales_country	nvarchar	50	是		销售国家

(3) 从商品类别维度表 (product_class 表, 见表 3.4) 中, 选取商品子类别、类别、部门、商品家族四个粒度。

表 3.4 商品类别维度表

属性名	类型	长度	能否 Null 值	主/外键	注释
product_class_id	Int	默认	否	P	商品类别 id
product_subcategory	nvarchar	50	是		商品子类
product_category	nvarchar	50	是		商品类
product_department	nvarchar	50	是		商品部门
product_family	Nvarchar	50	是		商品族

(4) 从部门维度表 (department 表, 见表 3.5) 中有市场、人力资源、会计等职位组成该维度中的粒度。

表 3.5 部门维度表

属性名	类型	长度	能否 Null 值	主/外键	注释
department_id	Int	默认	否	P	部门 id
department_description	nvarchar	100	是		部门详情

(5) 在销售事实表 (sales_fact_1997、sales_fact_1998 和 sales_fact_dec_1998 表设计, 见表 3.6) 中, 我们可以把销售量或销售金额作为指标。

表 3.6 销售事实表

属名	类型	长度	能否 Null 值	主/外键	注释
product_id	Int	默认	是	P	商品 id
time_id	int	默认	是		时间 id
region_id	int	默认	是		地区 id
department_id	int	默认	是		部门 id
store_sales	money	默认	是		商场销售价
store_cost	money	默认	是		商场成本价

unit_sales	float	默认	是	销售数量
------------	-------	----	---	------

第五节 工具开发环境设计

3.5.1 Java 开发与 MVC 模式

在 Flash analysis 的开发中我们采用 MVC 模式，由模型层（Model）、控制器（Controller）与视图层（View）组成。我们设计的模型层结构图 3.5 如下所示：

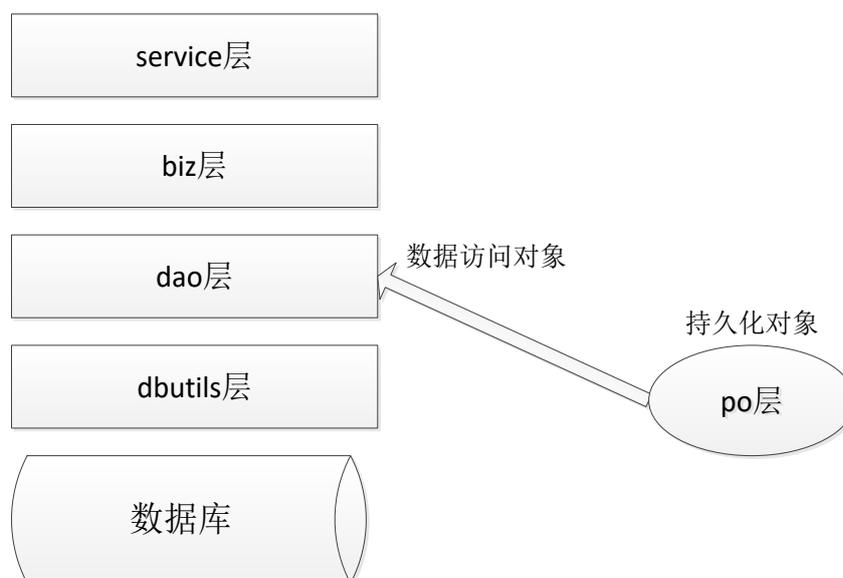


图 3.5 Data graph 的 MVC 框架模型层设计图

模型层里，db 层含 DBUtils 工具、connectionManager 连接以及 DBConfig 数据库配置，完成 Foodmart 数据库与 Java 连接的四个类。po 层是把底层数据库中的 Foodmart 表封装成 Java 中的实体类。dao 层进行对数据的查询访问，编写 sql 语句。biz 即业务逻辑层，service 是网络服务层。

在控制器中，Servlet 包实现模型层与视图层的连接，数据传输的关键^[14]。

最上层的视图层是系统的浏览器界面，我们在这里基于浏览器/服务器（B/S）模式，通过 jsp 实现数据的读入与回显。

3.5.2 操作系统与浏览器

我们是在 Windows7 版本以上的操作系统中开发的整个 Flash analysis 平台，Windows 系统是计算机中的主流系统，能够满足大部分用户。我们测试用的浏览器包括现在主流的 Google、360、IE9 以上的浏览器，这些浏览器能够很好的加

强数据可视化图形效果。

3.5.3 软件安装

在整个项目的开发过程中，我们用到了 MySQL 5.1 来存储用户的信息。MySQL 具有处理效率高，体积小，兼容性强，交互性能良好等优点，可以支持 Java、C++ 多种开发语言。其官网是 <http://www.mysql.com/downloads/>。

SQL Server 广泛的用在决策支持的多维数据联机分析处理上中，它集数据库与数据分析于一体，深受现代商业 Web 应用程序开发者青睐^[15]。SQL Server 的官方介绍网址是 <https://www.microsoft.com/en-us/server-cloud/products/sql-server/>。

我们系统的开发语言是 Java 语言，它是一种可移植性高、强健稳定、安全和简单的面向对象的多线程动态语言，它自带动态 Web 项目的创建，我们把 Eclipse 作为 Java 程序的开发工具能提高我们的效率。其官方下载地址是 <http://www.eclipse.org/downloads/>。

由于我们的 Flash analysis 系统是在 Web 浏览器中展现的，所以选择 Apache Tomcat 7.0.55 作为服务器。其兼具可扩展性能与安全性，并且在 Eclipse 中配置 Tomcat 服务器步骤也较简单。它的下载地址是 <http://tomcat.apache.org/>。

我们利用 D3.js 框架图形化显示数据，它的安装使用方法有两类，一类是在 D3 的官方中下载 d3.zip 文件，将其解压在项目前端 jsp 中的 js 文件下；另一类是在连接网络环境下，直接访问 d3.js 的网络地址。为了开发方便，我们采用第一类方法使用 D3。

3.5.4 外部包下载

用 Eclipse 开发 Flash analysis 系统过程中我们使用了很多 Java 外部包，来实现 Java 与数据库链接和访问、JSON 数组生成、用户头像上传。在 Eclipse 中，通过 Configure Build Path 可以导入外部 JAR 包，这些外部包可以在官网上免费下载获得。

与 MySQL 中的数据库连接中，我们用了 mysql-connector-java-5.0.8-bin.jar 以及 commons-collections-3.1.jar；为了能够与 SQL Server2008 连接，我们引入

了 sqljdbc4.jar。

在服务器的 servlet 层与前端 jsp 页面的数据传输过程中,我们将数据存储为 JSON 数组,目的是把封装的数据对象成 JSON 字符串,体积压缩以及便于读取。因此我们为了生成 JSON 数组,下载了 gson-2.2.4.jar。

在注册界面,我们为用户提供了头像上传功能,可以让用户在该系统中个性化地设计自己的形象。编写 Java 代码完成文件上传中也需要下载第三方提供的 fileupload 和 io 两个 jar 包。

第四章 数据封装与调用

第一节 数据请求响应

整个项目的视图层中，即读取数据与返回图形的前端浏览器界面，jsp 页面接受用户选择的信息。

在选择图形类型的界面，把四类图形的代表性图片展现在界面中，通过用户点击图片，链接到该图形的生成页面，每类图形对应一个 Servlet。

当用户进入图形获取页面时，先选定多维数据的维度、粒度和指标。我们在多维数据模型设计章节中，定义了四个维度，每个维度包含粒度，在前台页面每组维度采用联动的下拉框，jsp 获取联动框数据的 JavaScript 关键代码如下：

```
<script>
  var op=["op1", "op2"];
  var opt1=["dimension", "granularity"];
  function option(){ for (i=0; i<op.length-1;i++)
    document.getElementById(op[i].onChange=new Function("change("+i+1)"));
  change(0);}
</script>
```

用户在图形转换的一组单选框中选择一个图形后，其数据请求响应与下拉框相同。

在 Servlet 中，由于 doPost 能够输送大规模的数据信息，因此我们在 jsp 的 <form></form> 表单中采用 post 方法。二者的关键代码如下：

```
//Servlet 的 doPost 方法实现：
RequestDispatcher dispatcher = request.getRequestDispatcher("jsp/line.jsp");
dispatcher.forward(request, response);
```

```
//jsp 的 doPost 方法实现：
<form name="form1" method="post" action="">
</form>
```

第二节 数据查找

我们在 MVC 的模型层中创建一个专门用来连接数据库的包，Java 包中定义了连接管理类。由于 SQL Server 和 MySQL 数据库的连接外部 JAR 不同，因此二者的连接代码少有不同，MySQL 的连接代码如下：

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
return DriverManager.getConnection("jdbc:mysql://" + DBConfig.IP
+ ":" + DBConfig.PORT + "/" + DBConfig.DBNAME,
DBConfig.ACCOUNT, DBConfig.PASSWORD);
```

下面是 Java 与 SQL Server 2008 的连接关键代码，userName 和 userPwd 分别是 SQL Server 的用户名和登录密码：

```
Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver").newInstance();
String url = "jdbc:sqlserver://localhost:1433;databaseName=FoodMart 2005";
String userName="sa";
String userPwd="11111111";
return DriverManager.getConnection(url,userName,userPwd);
```

当 jsp 获取到用户的请求数据之后，模型层的 dao 层响应请求，启动与数据库的连接并开始查询工作。在 dao 层接口实现类中，首先需要定义连接类，以销售维度的 dao 层关键的 Java 代码为例：

```
private ConnectionManager2 connectionManager2;
private DBUtils dbUtils;
public SalesDaoImpl() {
super();
// TODO Auto-generated constructor stub
this.connectionManager2 = new ConnectionManager2();
this.dbUtils = new DBUtils();
}
```

之后开始 dao 层的数据查询，在查询之前我们需要有一个储存结果的空集合，下一步取得具体的数据库连接对象，接着便开始建立输写 sql 语句的模版，然后再用之前定义的 DBUtils 方法进行查询工作，随后创建具体的对象，最后把完成封装的对象添加到空集合里，完成数据的对应。我们以销售维度的 dao 层的接口实现类为例子，展示代码的关键部分：

```

public List<Sales> selectSales() {
    // TODO Auto-generated method stub
    // 步骤 1: 创建一个空的集合准备存放查询的结果
    List<Sales> lstSales= new ArrayList<Sales>();
    // 步骤 2: 获取一个数据库的连接对象
    Connection conn = this.connectionManager2.openConnection();
    // 步骤 3: 创建查询语句的模板
    String strSQL = "select the_date,count(unit_sales) from sales_fact_1997 as
A,product_class as B,time_by_day as C where A.time_id=B.time_id and
A.product_id=B.product_id and product_family='Drink' group by the_date";
    // 步骤 4: 使用 dbutils 方法实现查询操作
    ResultSet resultSet = this.dbUtils.executeQuery(conn, strSQL,new Object[] {});
    try {
        while (resultSet.next()) {
            // 步骤 5-1: 创建一个 Sales 对象
            Sales sales = new Sales();
            sales.setTheDate(resultSet.getDate(1));
            sales.setCount(resultSet.getInt(2));
            // 步骤 5-2: 将封装好的对象添加到 List<Sales>集合中
            lstSales.add(sales); }
        // 返回结果
        return lstSales;
    }
}

```

代码中的 sql 语句是表示找出 1997 年饮料类的商品每一天的销售总量，这里包含了时间维度和商品维度，指标是饮料类的每一天销量。

第三节 数据封装与转换

在 po 层我们把数据库中的对象与关系实体持久化的存储起来，将 Java 中的 dao 层数据与数据库的进行关联。在 po 层中我们了创建个五个类，分别是销售类（包括 1997 和 1998 年），商品类，人事类，时间类和地区类，与我们建立的多维数据模型中的四类维度和一类指标对应起来。以商品类和销售类为例，在商品类中，我们定义了商品种类 id，商品，商品类型和销售总数等私有成员变量。在销售类中定义了商场地址，销售总数，时间，单个销售数量等私有变量。po 层通过用 Serializable 来实现序列化功能。下面展示了销售类中的序列化代码：

```
@SuppressWarnings("serial")
public class Sales implements Serializable {
    private String store_state;
    private int count;
    private Date the_date;
    private int unit_sales;
    .....
}
```

在 po 层我们生成每个成员变量的 getter 和 setter 方法，并且创建带参函数。为了能够把持续化的对象转换成字符串，我们了生成 toString 方法。

我们把 dao 层分为 dao 层的接口方法和 dao 层的接口实现方法。在 dao 层的接口中我们定义了抽象方法类。biz 层我们同样分为抽象接口类和接口实现类，来调用 dao 层中的函数。

经过以上步骤我们已经完成了对数据对象的封装，但需要考虑的另一个问题是，到底怎样才能能在函数中轻松地输送这些数据对象呢？我们应当考虑把这些对象中表示的数据都转换为字符串，这时候我们就可以选择 JSON 来完成这项工作。它是一个轻量级的能够很容易地把数据传递的格式，它实现数据在 Java 各函数间以及前端与服务器端之中快捷、顺利地传递^[16]。在 D3 这类 JavaScript 框架中，JSON 格式的文件很容易被读取和解析^[17]。除此之外，它可以轻松地表达出十分难以解析的结构。正因为不论是对开发者还是计算机而言，都能不费力读懂、编码、解释和生成 JSON，所以它已经被众多程序编写者当作理想的数据传送结构。

选择在 Servlet 层编写获取 JSON 数组的代码，然后把获取到的数据轻松地传递到 jsp 中的 D3 框架中。现在的 D3 数据读取技术中，主流的方法是读取 CSV，TSV 和 JSON 等文件，另外如果用 D3 直接来读取数组，当数据量变庞大的时候，数据会占用很大的内存空间，而如果我们把 JSON 数组存放 JSON 文件中，文件会被存在硬盘里而非内存中，于是可以有效地保证系统的效率。所以我们需要引入下列三个包来分别用于 JSON 数据结构的生成和 Java 文件写入：`com.google.gson.Gson`；`com.google.gson.GsonBuilder`；`java.io.Writer`。我们通过下面代码实现了 JSON 数组的生成，先定义一个 Gson 对象，设定时间的格式为“yyyy-MM-dd”，然后用 toJson()方法生成 JSON 数组：

```
request.setCharacterEncoding("UTF-8");
HttpSession session = request.getSession();
Users user = (Users)session.getAttribute("user");
ISalesBiz salesBiz = new SalesBizImpl();
List<Sales> lstSales = salesBiz.selectSales();
session.setAttribute("user", user);
Gson gson= new GsonBuilder().setDateFormat("yyyy-MM-dd").create();
String s=gson.toJson(lstSales);
```

把 JSON 读入指定的 JSON 文件时，我们首先要定义好 JSON 文件的存储位置，再用 Java 中的 `Writer` 类把字符串写入 JSON 文件中，关键代码如下：

```
File file = new File("F:/BI-OLAP/json/ product.json"), dataList;
Writer write =new FileWriter(file);
write.write(s.toString());
write.flush();
write.close();
```

第四节 数据读取与映射

4.4.1 JSON 与 CSV 比较

获得的 JSON 文件中数据的存储格式就是通过使用 JavaScript 的数据对象的语法来创建字符串数组的格式，因此，JSON 这类的数据比其他的数据格式例如 CSV，更容易被 JavaScript 语言解析。如今很多使用 D3 技术开发的数据可视化图形，都是用 CSV 存储数据，然而，D3 对 CSV 文件中的数据格式要求很高，结构严谨，有时候需要 CSV 用逗号分隔属性，有时候却只能解析用空格分隔属性的 CVS。但是，在 JSON 文件中就不需要考虑太多这类问题，D3 能够很灵活地处理 JSON 文件。所以用 JSON 来存放数据要比用 CSV 更好。

4.4.2 JSON 数据格式

JSON 可以很容易地把一个数据对象转变为一个字符串的形式，在属性和值之间用冒号隔开，逗号将两对属性和值隔开，大括号将两组对象分隔。于是我们得到的 JSON 文件中数据格式如下：

```
[{"count":117,"the_date":"1997-01-01","unit_sales":0},
{"count":128,"the_date":"1997-01-03","unit_sales":0},
{"count":13,"the_date":"1997-01-04","unit_sales":0},
{"count":127,"the_date":"1997-01-05","unit_sales":0},
.....
]
```

4.4.3 D3 的数据读取与映射

D3 框架中使用 `d3.json` 专门读取 JSON 文件，把数据读入 `data` 中，`domain` 方法用来实现横纵坐标，调用 `function(d)` 来映射 JSON 文件中数据的属性和值，关键代码如下。

```
d3.json("/BI-OLAP/json/produnct.json", function(error, data) {
  x.domain(data.map(function(d) { return d.testfamily; }));
  y.domain([0, d3.max(data, function(d) { return d.count; })]);
});
```

与读取 CSV 文件的 `d3.csv` 不同，`d3.json` 括号中除了包含文件的 `url` 和 `function(error,data)` 之外，不能再含有其他定义数据格式的函数，例如 `function type(d)`，这一函数常常用来将某一属性中的值封装转换为在 `var` 中定义的格式。但在读 `json` 文件时，不能使用 `type`，而需要在每一次调用这个属性的值时，在前面都加上 `var` 中定义的格式。例如我们在画折线图时，以时间为横坐标，销量数为纵坐标，我们把时间格式统一规定为 `"%Y-%m-%d"`，这部分的关键代码如下：

```
var formatDate = d3.time.format("%Y-%m-%d");
var line = d3.svg.line()
.x(function(d) { return x(formatDate.parse(d.the_date)); })
.y(function(d) { return y(d.count); });
```

第五章 Data graph 工具实现

第一节 交互选择功能实现

在这一小节中，我们主要介绍系统首页面展示的图形选择功能和图形生成页面的维度选择功能。用户在整个系统首页访问 Data graph 工具五个图形模块，最上的是热点图，下方从左到右分别是柱、条、饼状图和地图，用图片而非文字的连接，目的是为用户提供最直观的选择方式，界面见图 5.1。

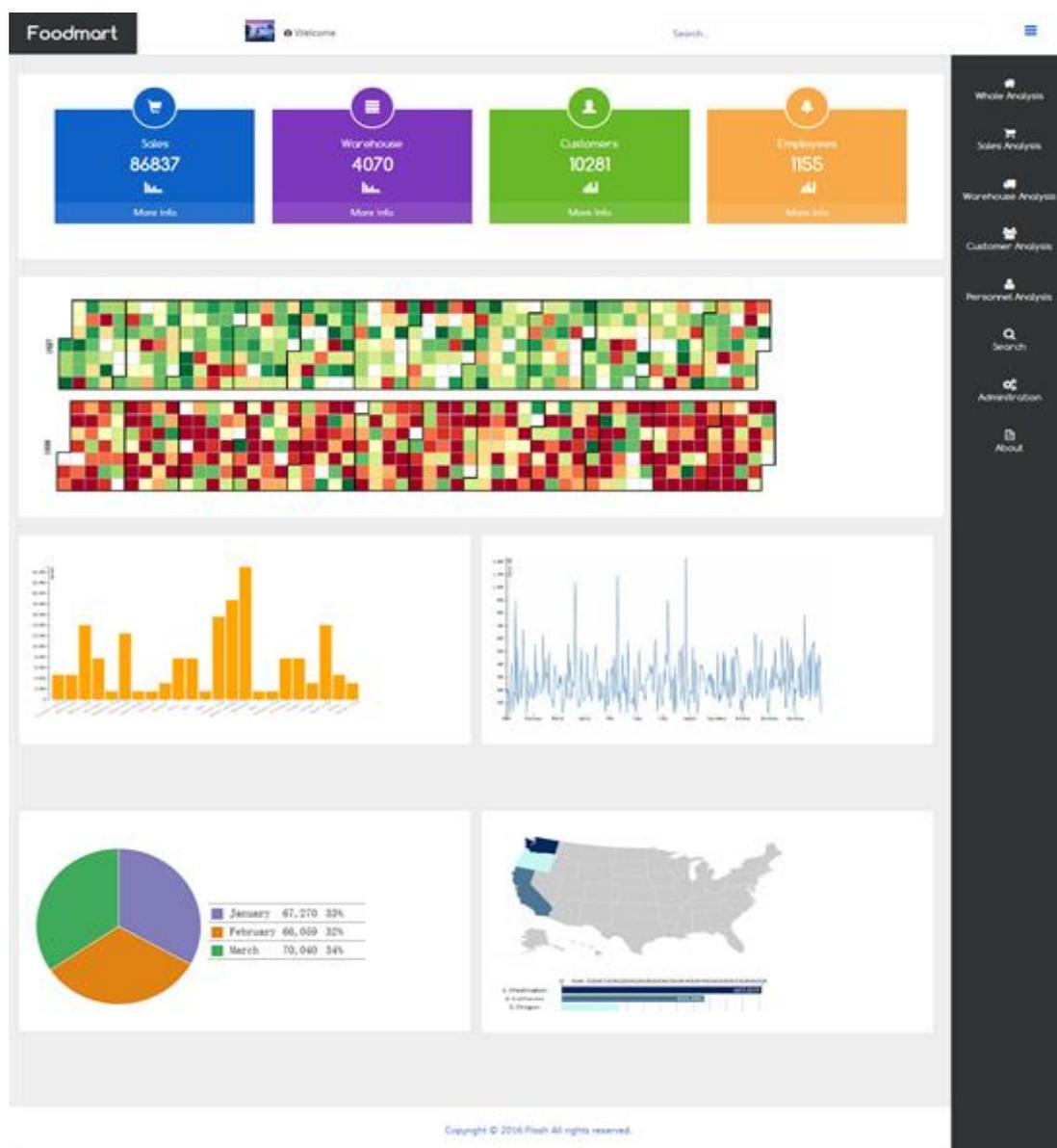


图 5.1 Flash analysis 系统首页

当用户点击图片后，就可以进入对应图形的生成界面，该界面以白色为主，色彩简单，可以衬托出图形结果的色彩。所有图形的生成界面其组成部分都包括四组独立的维度、粒度选择联动框，销售数量和销售金额确认按钮，一个图形转换下拉框，相应图形的示例图和对示例图的颜色表达含义，坐标表示以及使用方法的详细文字说明。我们以热点图为例子，展示图形生成界面的效果，见图 5.2。四个下拉框中从上至下分别是：Time Analysis（时间分析），Region Analysis（地区分析），Product Analysis（产品分析）和 Personnel Analysis（人事分析）。

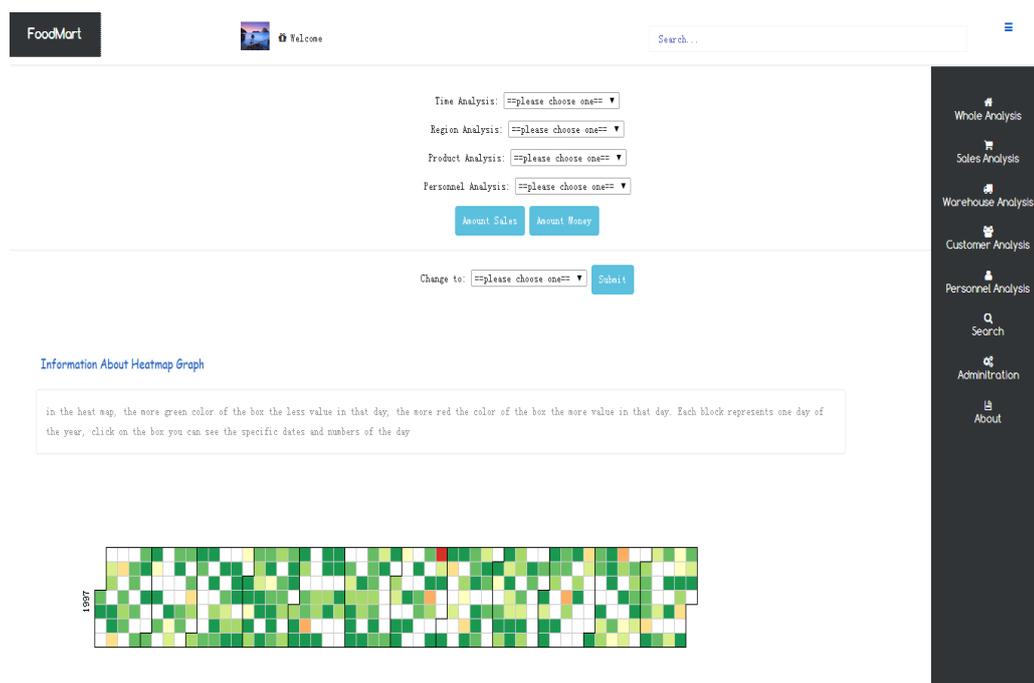


图 5.2 图形生成页面

例如，用户在时间维度中选择了 1997 年中的日粒度，在商品维度中选择了商品族中食品粒度，点击销售量按钮，并且希望得到相应的折线图进行对比。此时联动下拉框的显示结果如下图 5.3 所示。

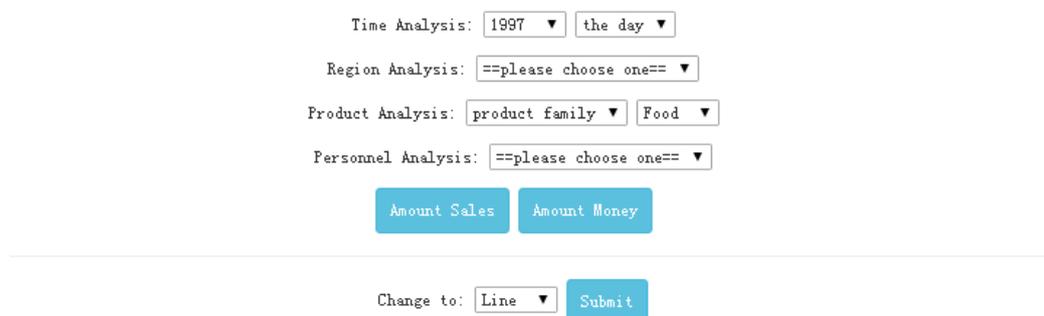


图 5.3 联动下拉框

第二节 传统图形实现

第二小节我们主要展示 Data graph 工具的传统图形模块中三类图形的 D3 技术实现。D3.js 可以通过用 HTML 或者 SVG 将数据映射到图形上，在这个工具开发过程中，我们是采用 SVG 格式，它可以构建二维图形，并且实现交互、动画功能^[18]。

5.2.1 柱形图

柱形图是十分常见的、基本的二维图形，其特点是可以映射多种类型的数据。柱形图的基本框架，包括水平轴（横轴，即 X 轴）、垂直轴（纵轴，即 Y 轴）和柱形。

柱状图的竖轴不仅可以代表类别数据和时间变化的数据。在表示时间变化的时候，我们可以称其为时间轴，时间顺序从左到右，可以表示年、月、日等各种时间维度上的粒度。当横坐标用来表示类别时，一个柱形表示一类。垂直轴可以用来显示数值，以标出图形的数量范围，每条柱形高度的分别与左侧的垂直轴一一呼应，柱形的高度便代表了数值的大小，例如，矮的柱形代表的数值比高的柱形代表的数值小。除此之外，我们还要注意每个柱形的宽度和间隔。有些时候，柱形不同的宽度可以代表数值的范围大小^[19]。

画出柱形图的横纵坐标时，我们可以用 var 先定义两个变量分别代表横纵坐标，调用 d3.svg.axis() 函数，使用 orient 插件可以设置方向：

```
var xAxis = d3.svg.axis()
    .scale(x)
    .orient("bottom");
var yAxis = d3.svg.axis()
    .scale(y)
    .orient("left");
```

在实现了基本的柱形图框架之后，我们可以在朴素的柱形图上添加一些交互效果，能够让传统的柱形图变得生动起来。当用户把鼠标停留在到某一个具体的柱形图上时，柱形图由原来的橙黄色动态标为鲜红色，并且出现悬浮框，标有对应水平坐标的值和数值轴的值。我们可以先定义一个 tip 变量，调用 d3.tip() 元素，

在 HTML 的函数返回值中使用 `` 标签展现红色，在把两个二维坐标的值抽取出来，通过 `` 标签来加强文字的显示效果，这部分的关键代码如下：

```
var tip = d3.tip()
  .attr('class', 'd3-tip')
  .offset([-10, 0])
  .html(function(d) {
    return "<strong>Type:</strong> <span style='color:red'>" + d.testfamily +
      "</span><br>" +
      "<strong>Amount:</strong> <span style='color:red'>" + d.count + "</span>";
  })
```

在 `svg` 中可以把整个柱形图的宽度与高度对应起来，之后再使用 `svg.call(tip)` 激活鼠标悬停在柱形上的交互效果，用 `svg` 画出柱形图的关键代码如下：

```
var svg = d3.select("body").append("svg")
  .attr("width", width + margin.left + margin.right)
  .attr("height", height + margin.top + margin.bottom)
  .append("g")
  .attr("transform", "translate(" + margin.left + "," + margin.top + ")");
svg.call(tip);
```

我们选择时间维度上的 1997 年里的每周周日，商品维度中商品族里的食物类，查看每类食物销量的柱形图为例，展示 Data graph 工具中该类图形的模型，见图 5.4 所示。横坐标是食物类中的商品分类，共有 15 类，由于是按照字母顺序排列，所以最左边是烘烤食物（Baked Goods），最右边是淀粉食物（Starchy Foods）。纵坐标是各类食物整个一年的周日销售的总量。柱形图的横轴宽度是固定的，因此柱形的宽度会随着水平轴的类别个数自适应调整，每个柱形的宽度都相同。我们可以从图中获得如下信息：在周日这一天中，果蔬等农产品（Produce）的销量最多，有可能是因为周末消费者数量增大，同时也可能是价格便宜消费者最喜欢购买果蔬等农产品。因此商家可以按照这个历史值在以后的周日准备充足的果蔬等农产品，以满足顾客的需求，针对卖出量很少的肉类和海鲜类，商家可以考虑做活动来推广产品，以提高销量。在顾客角度可以通过这个柱形图看出，由于购买人数很多想要购买到新鲜的果蔬农产品在周日这一天要选择尽量早的时间购买。

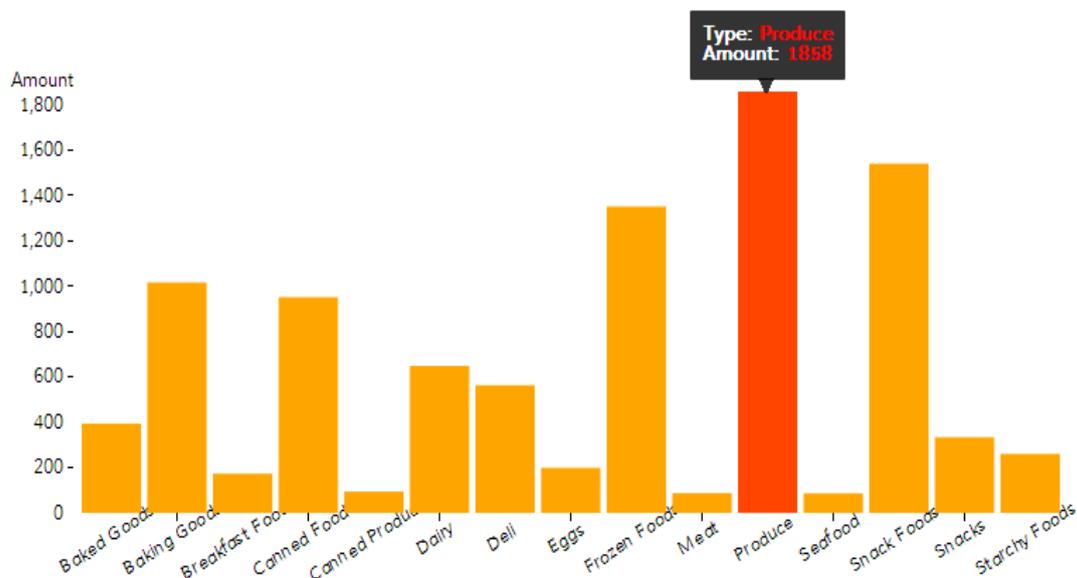


图 5.4 Data graph 柱形图

5.2.2 折线图

折线图的基本组件也包括了有横纵坐标，折线图的主体部分是随着数值的大小而弯曲变化的曲线。与柱形图不同，折线图不能够表示离散的数据，水平方向上的轴代表的是持续性地维度，比如时间维度^[19]。因为，折线图能够很好表示某类事物的时间变化趋势。曲线的波动越大，代表在某个连续的时间段内，数值的变化越大。与所有包含横纵坐标的图形相同，在 d3.js 技术中，用 d3.json 读取数据之后，需要用 x.domain 与 y.domain 插件映射横纵坐标的代表的数据维度。通常情况下，横坐标的值很多，而文字很长，排版会十分密集，导致显示不清晰，这时候我们用 attr() 插件中的 rotate()，来改变文字的倾斜度，以错开相邻文字的距离，例如 rotate(0) 代表与竖直方向平行，负号代表逆时针方向旋转，而正号则表示向顺时针方向旋转。我们用 text() 插件来显示坐标的标题。我们通过 svg.append 将 data 中存放的数据映射到折线上，用一下代码：

```
svg.append("path")
  .datum(data)
  .attr("class", "line")
  .attr("d", line);
```

如果我们选择时间维度上的 1997 年 1 月至 12 月，商品维度上的商品子类的

酸奶，查看每天的销售量的折线图，结果如图 5.5 所示：

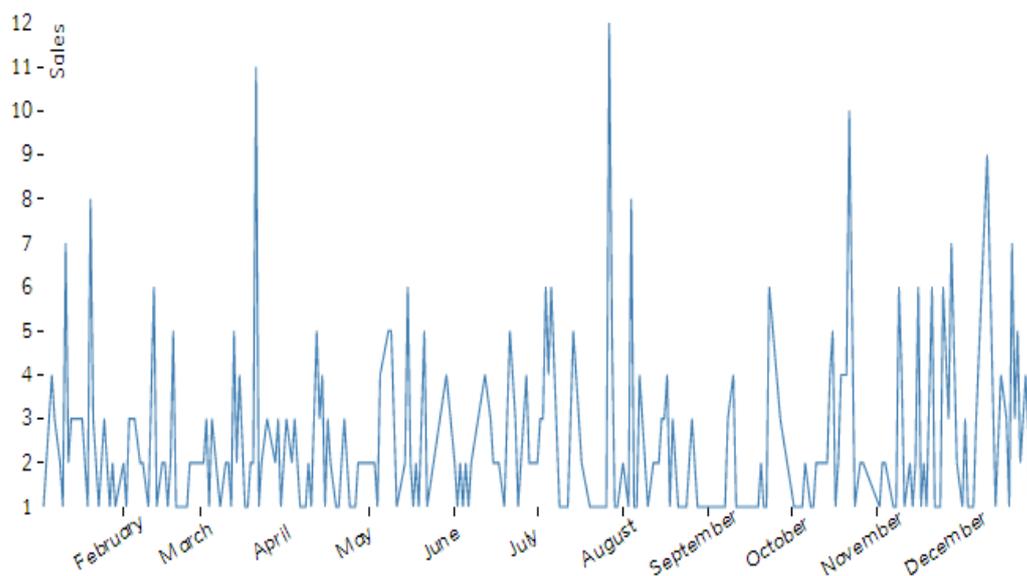


图 5.5 Data graph 折线图

在 Data graph 工具中所有的折线都是用蓝色细线显示。从上图我们能够看出，横坐标显示的时间段从 1 月延长到 12 月，下标是倾斜排列，可以避免密集造成显示效果不佳。垂直坐标最上方表有销售 (sales)，表明是每天卖出量。折线图的趋势表明，全年当中的 11、12 月份两个月平均销量较大，可能是因为 11、12 月中的感恩节和圣诞节，顾客对酸奶的需求量很大。在 7 月某一天销售量达到最大值，而在 8、9 月份的销售量很小。这有可能是在 8、9 酸奶的生产量较，或者就是因为销售量小。商家可以根据这个折线图做一个市场调查，如果是因为顾客在这两个月普遍消费不大，那么商场可以在这两个月中做酸奶打折促销活动，来带动顾客的购买欲望，以提高全年的总销量。

5.2.3 饼图

和前两种图形有所不同的，饼图主要特点是能够表达出数据之间的比例关系。我们往往希望能够直观地获得某一类或者某一子类事物中整体的分布、数量最多的、最少的。而然数量上最多、最少可以用柱状和折线表示，但饼图更关注于事物之间的占比^[19]。

在绘制饼图的时候，圆的半径是主要元素。在 d3.js 中用 d3.svg.arc() 插件来

画圆。innerRadius()和 outerRadius()分别用来定义圆的内半径和外半径，从而可以得到一个环状的饼图。我们可以通过 color 变量来实现不同颜色，Data graph 的饼图中可以表示 20 种不同颜色的占比，用 d3.scale.category20()来实现。在饼图中，需要用到<path>元素，称为路径元素，通过定义好的一段路径来画出饼图。关键代码如下：

```
var path = svg.datum(data).selectAll("path")
    .data(pie)
    .enter().append("path")
    .attr("fill", function(d, i) { return color(i); })
    .attr("d", arc);
```

如果用户在地区维度上选择了美国各州省的商业分布，查看饼图，结果如图 5.6 所示。商业分布在华盛顿州、加州和俄勒冈州，其中华盛顿州所占的数量最多。除此之外，饼图还能够广泛用在其它有关分类和比例结合的数据上。

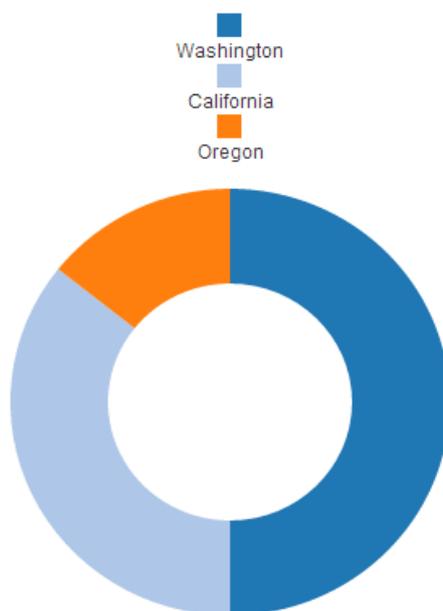


图 5.6 Data graph 饼图

在这一小节中我们介绍 Data graph 工具中的热点图和地图的实现过程。热点图和地图与传统的二维图形有所不同，它们可以比较一组数据中的多个变量。分析者可以从热点图中根据颜色的深浅找到异常值，从地图中可以分析空间上的差异。

第三节 热点图与地图实现

5.3.1 热点图

当用户需要在一个以上的变量之间做比较的时候，可以选择用热点图来进行分析。热点图的基本组成一般包含按行按列组合的一个个单元格。行可以代表一个对象或者一个分析角度，列可以代表对象的某一个变量。一个个排列在一起的颜色深、浅不同的单元格表示不同大小的数值。在热点图中，我们可以通过单元格中的颜色，很快就能发现某个较大或者较低的值。与普通的二维坐标的图形相比，热点图的布局有些复杂，但是其通过颜色深浅辨别，在某些场合还是能够为分析人员提供不小的帮助。

Data graph 工具的热点图可以用来展示一年中，每一天的数值。每个小方格代表一年里的一列代表一年里的某一周，每四周便代表一个月。每一行表示每一周中的星期几。所以该热点图可以将时间维度上的，年、季度、月、周、日和每月第几天、每年第几周、每年第几月都在一个图上显示出来，再根据颜色的不同，可以从不同角度做分析，可以达到事半功半的效果。在 `<style></style>` 标签中我们用 `.RdYlGn` 中的 `fill` 元素定义不同的颜色，需要多少种不同的颜色，就调用多少次 `.RdYlGn`。我们定义了绿色从深到浅，表示数值从小到大，红色从深到浅，表示数值从大到小，也就是说，深绿色表示的数值最小，深红色表示的数值最大。`color` 元素可以把一个区间内的不同大小数值映射到不同颜色上，`domain` 元素则用来规定能够表示的数值的区间。这部分的关键代码如下：

```
var color = d3.scale.quantize().domain([ 0, 400 ]).range(
    d3.range(11).map(function(d) {
        return "q" + d + "-11";
    }));
//domain([0,400]) 表示我们在这里定义颜色对应的数值区间在 0 到 400 之间
```

与柱形图的交互功能相似，在热点图中，我们也实现了信息显示的交互效果。当鼠标停留在每个单元格上，鼠标右侧将会出现一个白色的小框，显示单元格对应的日期和这一天中某个维度的数值。

我们以在时间维度选择 1997 年，在地区维度上选择加州（CA），然后查看

所有的商品销量为例子，展示 Data graph 工具的热点图的效果，如图 5.7 所示：



图 5.7 Data graph 热点图

在图中我们可以看出，在 1997 年的 10 月 19 日这一天的单元格中是深红色，表示销量最多，鼠标停留在该方格上，信息显示这一天加州所有商场的总销量是 381 个。从左到右的 12 个方格组表示 1 月到 12 月，我们可以看到，11 月和 12 月的销量很多，是消费者购买力最大的两个月。同样，从上至下，7 列依次代表从周日至周六，从整体上看，周日和周六的绿色颜色普遍较浅，红色颜色的小方格较多，由此粗略推断，周六和周日，加州商场的顾客购买力是最强的。

5.3.2 地图

地图是一种十分有助于直观地理解数据的数据可视化图形，从小孩到老人都很容易地读懂的含义，因此，在多维数据可视化表型形式中地图占了很重要的地位。地图能够把真实的地理分布按照比例缩放，将抽象的、庞大的世界清晰地展现给人们。在 Foodmart 数据库中，凡是有关空间的数据都可以用地图的形式映射出来。

在绘制地图上精确表示某个国家或城市的位置时，我们需要关注的是该地方的经度和纬度的数值，但是经度和纬度的数值十分复杂，我们就可以使用已有的 us.json 文件来画出含有各州分块的美国地图，这样可以大大减少我们的编程量，该文件的下载地址是 <https://s3-us-west-2.amazonaws.com/vida-public/geo/us.json>。然后通过 makeMap 和 makeBars 把地图上的城市和其对应的数据结合起来^[20]。在 Data graph 工具中，我们用不同的颜色表示数量的多少。也就是说，在柱形图中表示数据类别的横坐标在地图上用城市的地理分布来代替，而表示数值的纵坐标在地图上则用颜色来代替。这样我们可以非常直观地把每个地方在地理位置上的关系变量以及它们的数值关系变量都表达出来。

在生成地图界面，如果我们选择地区维度上的销售州省这个粒度，时间维度

上的 1998 年，图形的显示结果如图 5.8 所示：

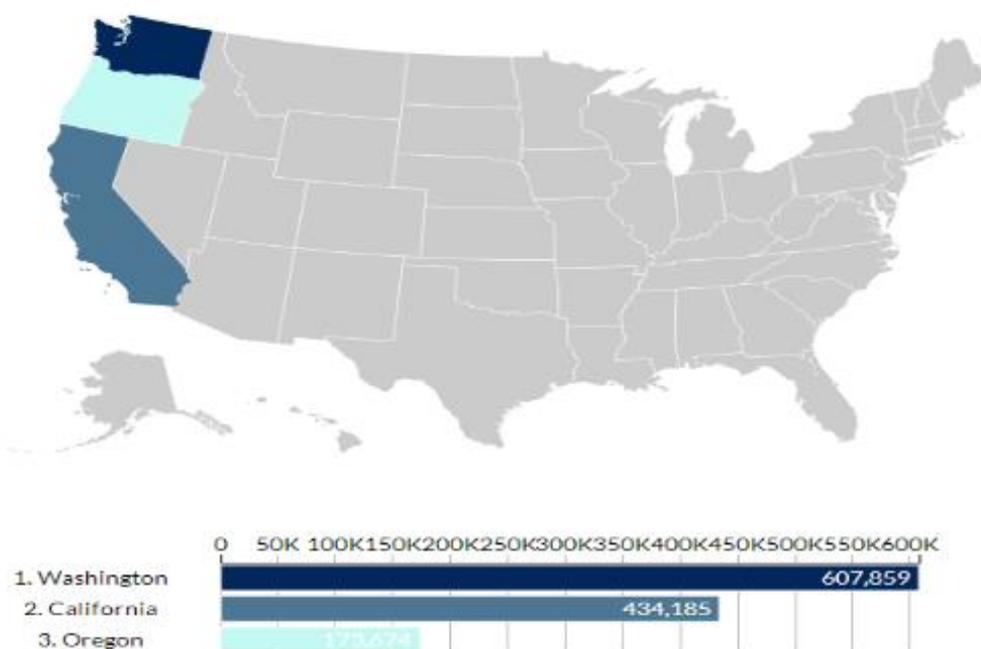


图 5.8 Data graph 地图

在 Data graph 工具的地图图形中，我们用不同深浅的蓝色来表示数量的大小，蓝色颜色越深，表示的数值越大，相反则数值越小。从上图我们可以看到，Foodmart 数据库中记录的商场所分布的州省有华盛顿州、加州和俄勒冈州。最上方的华盛顿州蓝颜色最深，表示 1998 年里华盛顿州的销量最大，商场的投资者可以在华盛顿州进行多一些的业务拓展。商家也可以在消费量最少的俄勒冈州进行市场调查，看看该地区的顾客最喜欢的商品是什么，通过产品推广活动，来提高该地的购买力。

5.2.3 图形转换

图形的转换功能目的是将同一组数据信息通过不同的图形展现出来，所以用户可以从不同的角度深入分析这组数据的关联信息。在 Data graph 工具里，用户可以选择连续一年时间段里的数据，然后进行柱形图、折线图与热点图的转换；离散的数据可以在柱形图和饼图中转换，有关地理的数据还可以转换成地图。

第四节 系统其他工具展示

在这一节中我们详细展示 Flash analysis 系统中除 Data graph 工具之外其他几个部分的界面。包括注册、登录、系统首界面以及多维数据联机分析和数据查询工具的界面。

5.4.1 注册界面

如图 5.9 所示,初次使用 Flash analysis 系统的用户需要在注册页面填写昵称、邮箱、密码信息,点击选择文件按钮,上传头像,并且同意系统的使用条款,点击 Sign up 确认注册,已注册的用户可以点击 Login 按钮连接到登录界面。

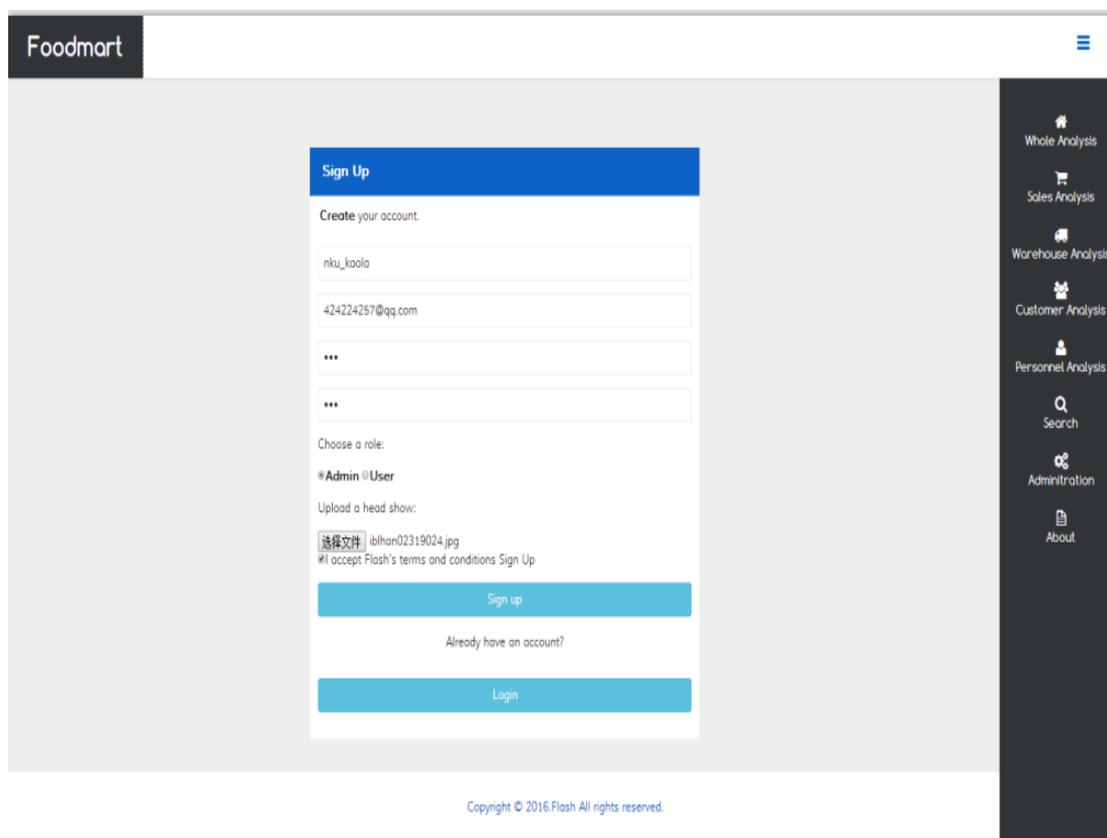


图 5.9 注册界面

5.4.2 登录界面

新用户注册成功后,系统自动跳转到登录页面,见图 5.10,这时候登录界面的昵称输入框会自动显示新用户注册时使用的昵称。用户只需再输入密码点击

Sign in 按钮就能链接到 Flash analysis 系统的首页。

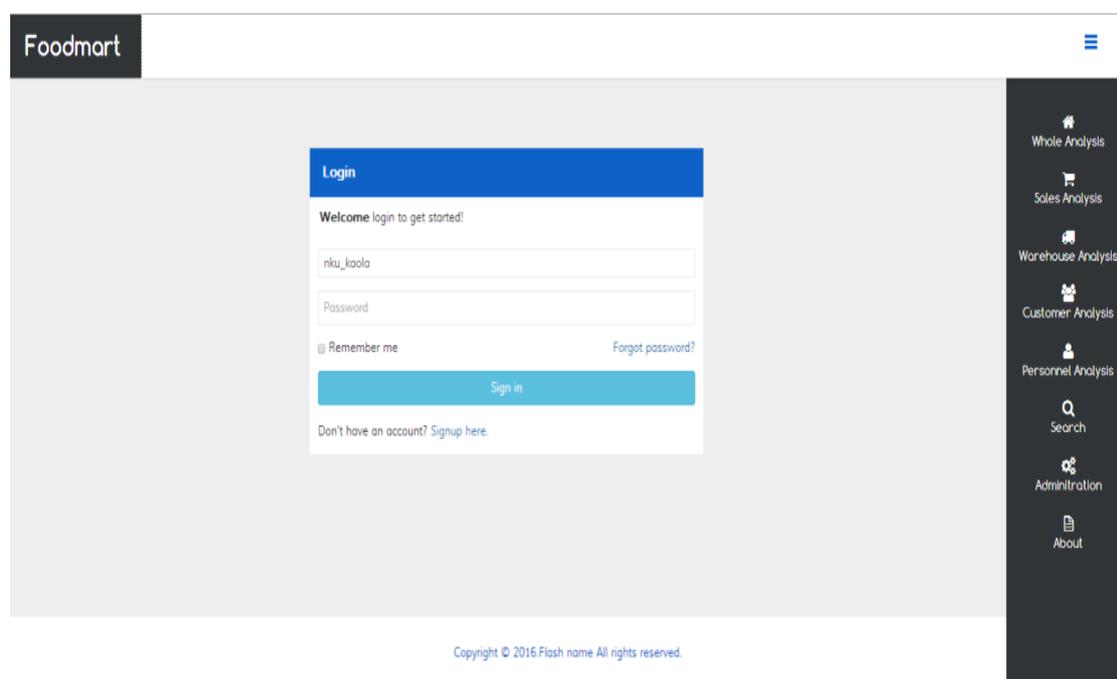


图 5.10 登录界面

5.4.3 主界面

Flash analysis 的主界面包括四个部分。页面的最上方白色导航栏区域最左边是系统的黑白色 logo，写着“Foodmart”，右边相邻的部分是用户头像，搜索框。点击用户的头像会出现一个下拉菜单，第一行是用户名，第二行是退出登录按钮。头像右边是“Welcome”字样，表示欢迎用户的登录。搜索框可以进行模糊信息查询。首页的右侧黑色菜单栏部分连接到八个功能模块，最上方整体分析（Whole Analysis）连接到首页，下面四个分别是多维数据的联机分析的四个分析模块即销售分析（Sales Analysis）、仓库分析（Warehouse Analysis）、客户分析（Customer Analysis）和人力资源分析（Personnel Analysis），接着是信息查询（Search）和系统介绍页面（About）。点击菜单栏上的这些文字，可以进入到对应的界面中。首页面的中间部分又分为上下两个部分。上部分是四个颜色不同的卡片似的标签，标签上显示有四个不同的数据，从左到右是 Foodmart 数据库中销售总量、仓库货物总量、顾客总数和员工总数，这一部分是从总体上展示 Foodmart 数据库中的信息，由此可以看出在整个 Foodmart 数据库中记录的信息量很大。下半部分是 Data graph 的图形选择部分。Flash analysis 系统的首页主要特点是图文结

合，为了能够让用户使用方便并且增强网站的美观，我们在很多地方使用了小图标与相应的文字搭配起来。系统的首页完整布局见图 5.6，为了能够清楚展示细节部分，我们又截取了首页的上半部分，见图 5.11。

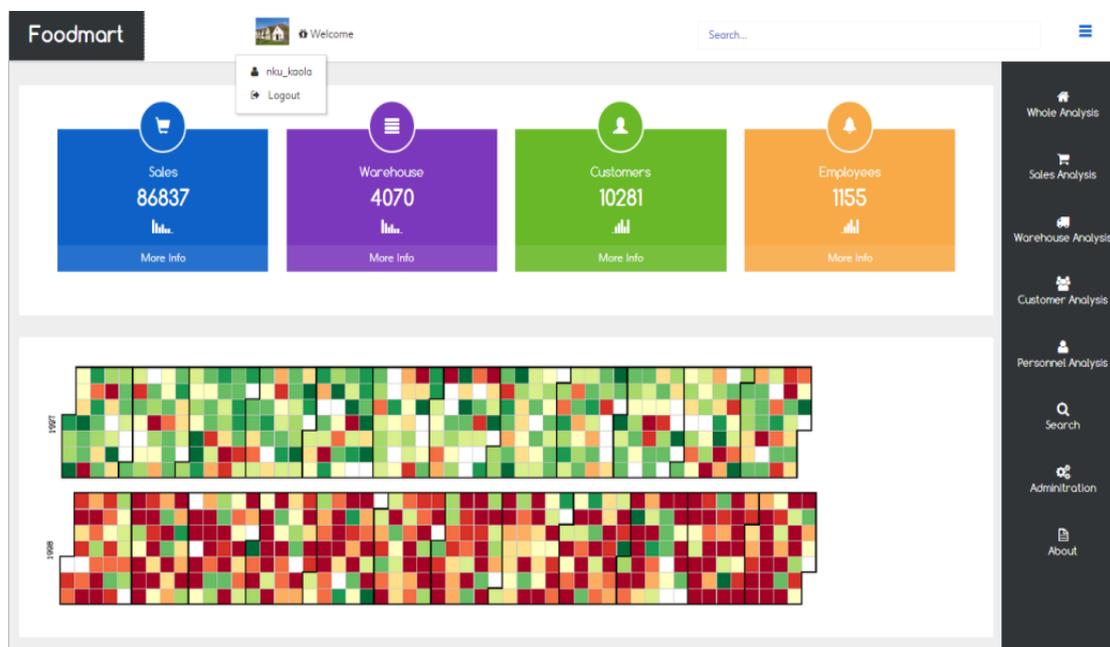


图 5.11 Data analysis 首页上半部分

5.4.4 多维数据联机分析界面

在 Flash analysis 系统中多维数据联机分析工具是使用了 Mondrian 模式，利用 Jpivot 标签库绘制联机分析处理的表格和图标，并提供报表的 Excel 文件下载。见图 5.12，多维数据联机分析中生成的图形只包括普通的柱形、饼形和折线图，且图形中没有鼠标交互功能，Data graph 工具可以很好的弥补了它的不足之处。

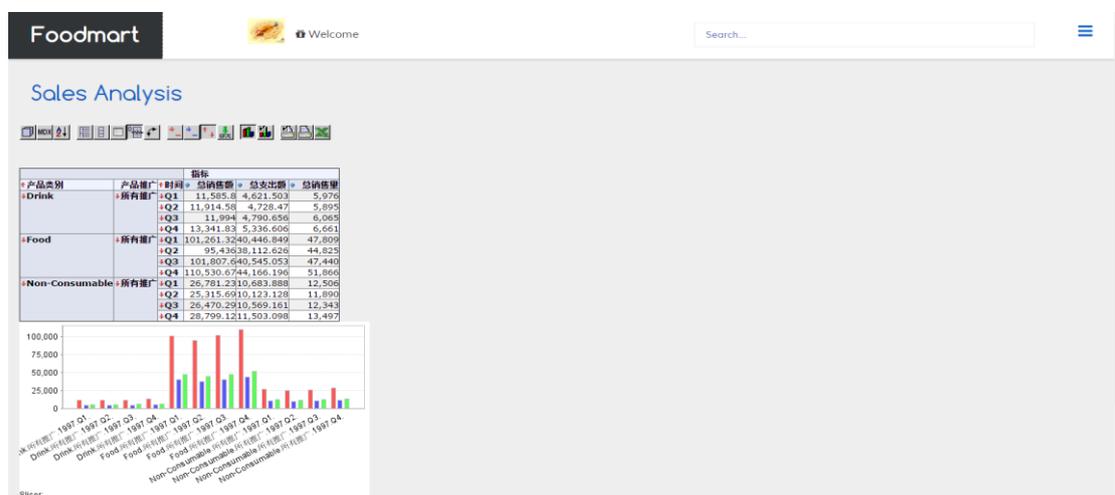


图 5.12 多维数据联机处理分析页面

5.4.5 数据查询界面

在数据查询界面，用户可以通过模糊搜索信息，可以查找有关的汇率，部门介绍和品牌销售量等数据，以表格的形式展现出来。数据查询界面如图 5.13 所示：

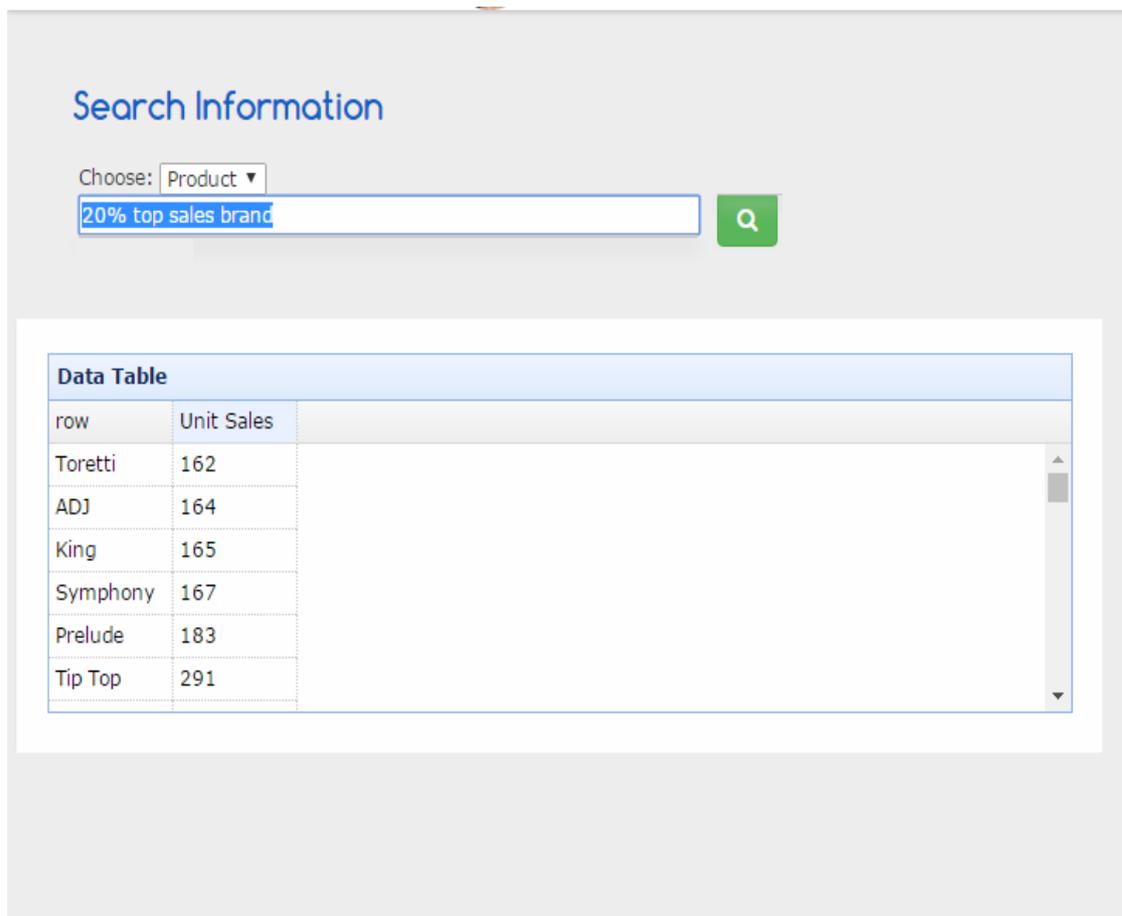


图 5.13 数据查询面

第六章 总结与展望

第一节 总结

在这篇文章中，我们主要为 Flash analysis 商务智能系统设计与实现了一个多维数据可视化工具，即 Data graph 工具。这个工具主要为商业分析人员和一般的商场消费者生成直观、个性化的数据可视化图形，辅助他们进行商业决策和对未来市场的预测。工具为用户提供了有交互功能的柱形图和热点图，以及折线图、饼图和地图五种图形。用户只需要点击相应的图形预览图，进入生成图形界面，使用下拉框确定维度即可得到图形结果，用户还可以根据需要将不同的图形进行转换，从不同的视觉角度深入分析数据特点。

我们使用 Java 语言和 MVC 模式搭建工具的服务器端，以连接后台的 Foodmart 数据库和前端的浏览器界面。D3 技术实现了各类图形在浏览器中的显示，数据存储于 JSON 文件中，用来实现数据对象在服务器与前端的快速传递。

Data graph 工具的操作十分简单，只需要简单的鼠标点击就可以获得图形结果。每个图形的生成界面配以图形使用说明，便于用户读懂图形结果。工具的界面以白色为背景，结构紧凑。

Data graph 工具能够和系统中的多维数据联机处理工具、数据查询工具功能互补，使得整个 Flash analysis 系统功能齐全，可以全方位的为用户提供数据分析服务。

第二节 展望

我们基本实现了 Data graph 工具的预期设计的功能，但还有存在许多不足的、可扩展的地方。有以下几点：1) 我们希望能够增加更多的图形模块，例如双坐标柱形图，玫瑰图和散点图等；2) 提供更多的维度选择，例如仓库维度、顾客维度等，增加更多的功能；3) 目前我们开发的 Flash analysis 系统主要是服务于分析 Foodmart 数据库，我们希望能够将系统扩展成通用的商务智能系统，为不同的商务企业数据分析提供服务。

参考文献

- [1] Negash S. Business intelligence. *Studies in Computational Intelligence*, 2003, 13(5):177~195.
- [2] Luhn H P. A business intelligence system. *Ibm Journal of Research & Development*, 1958, 2(4):314~319.
- [3] 任磊, 杜一, 马帅等. 大数据可视分析综述. *软件学报*, 2014(9):1909~936.
- [4] 韩丽娜. 数据可视化技术及其应用展望. *煤矿现代化*, 2005(6):39~40.
- [5] Mcdonald L. Florence Nightingale, statistics and the Crimean War. *Journal of the Royal Statistical Society*, 2013, 177(3):569~586.
- [6] Neuhauser D. Florence Nightingale gets no respect: as a statistician that is. *Quality & Safety in Health Care*, 2003, 12(4):607~614.
- [7] 刘芳. 信息可视化技术及应用研究[博士学位论文]. 浙江大学, 2013.
- [8] 任永功, 于戈. 数据可视化技术的研究与进展. *计算机科学*, 2004, 31(12):92~96.
- [9] 赵倩. 基于可视化分析的交互式仪表盘系统的设计与实现[硕士学位论文]. 中国科学院软件研究所, 2009.
- [10] 黄性清, 柳清. ERP 商务智能系统的设计方案. *中国管理信息化*, 2011(21):47~50.
- [11] Michael B, Vadim O, Jeffrey H. D3: Data-Driven Documents. *IEEE Transactions on Visualization & Computer Graphics*, 2011, 17(12):2301~2309.
- [12] 王兴玲. SVG 与矢量地图的 Web 发布技术. *计算机工程与应用*, 2002, 38(10):1~4.
- [13] 高静, 段会川. JSON 数据传输效率研究. *计算机工程与设计*, 2011, 32(7):2267~2270.
- [14] 陆荣幸, 郁洲, 阮永良,等. J2EE 平台上 MVC 设计模式的研究与实现. *计算机应用研究*, 2003, 20(3):144~146.
- [15] 朱德利. SQL SERVER 2005 数据挖掘与商业智能完全解决方案(附光盘). 电子工业出版社, 2007.
- [16] Wang G. Improving Data Transmission in Web Applications via the Translation between XML and JSON. *International Conference on Communications and Mobile Computing*, Cmc 2011, Qingdao, China, 18-20 April. 2011:182~185.

- [17] Crockford D, Crockford D. The application/json Media Type for JavaScript Object Notation (JSON). *Journal of Biological Regulators & Homeostatic Agents*, 2006, 13(4):250~251.
- [18] 邱德清. 基于 web 的多维数据可视化工具的研究与实现[硕士学位论文]. 北京邮电大学, 2015.
- [19] Nathan Yau 著, 向怡宁 译. 鲜活的数据. 北京: 人民邮电出版社, 2012-10.
- [20] Zhu N Q. *Data visualization with D3.js cookbook*. Birmingham-Mumbai: Packt Publishing. 2013.

致 谢

2012年9月父母送我一起来到南开，踏进这个陌生又让人亲切的校园，当时对未来四年的生活又期盼但又有些害怕。这四年的时间，从未知到尘埃落定，从迷茫到从容，从八里台到津南，一边探索一边成长，有喜有忧，南开的精神一直在激励着我，让我成为更好的人。

毕业论文即将完成之际，我真诚地感谢谢茂强老师对我毕业设计的耐心教导和鼓励。谢老师一直以来都用严谨的教学态度、扎实的学术理论来引导着我们。每一次向谢老师汇报完毕业设计的进度，都能够获得很多的新知识，思维也被打开。在每一次汇报中，平易近人的谢老师让我学到了很多生活、学术中的道理，受益良多。

我要感谢毕业设计项目小组的李宛恒、李羿霏同学，谢谢你们这一段时间对我的无私帮助和包容。

再次，我要深深地感谢我的父母，谢谢你们的爱与信任，谢谢你们多年以来默默地付出。

最后感谢所有帮助我、支持我的同学、老师，你们的建议与关心将是照亮我今后学习道路的明灯！

朱晴宇

2016年4月于南开园