

本科毕业设计论文

题目：交通大数据网站的设计与开发

|  |  |
| --- | --- |
| **作者姓名**  **指导教师**  **专业班级**  **学 院** | **朱鑫栋** |
| **李永强 教授** |
| **电气工程及其自动化1401** |
| **信息工程学院** |

**提交日期** 2018年6月05日

**Dissertation Submitted to Zhejiang University of Technology**

**for the Degree of Bachelor**

**Dissertation Title (Times New Roman, Bold, Centered, Multi-lines allowed)**

**Student: Zhu Xindong**

**Advisor: Li Yongqiang**

**College of Information Engineering**

**Zhejiang University of Technology**

**June 20xx**

浙江工业大学

本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所呈交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

年 月 日

**本科毕业设计的任务书**

（用盖过红章的任务书替换）

交通大数据网站的设计与开发

摘 要

现代化的智能交通系统往往以大数据为基础，不断分析大量的用户、设备数据，从中找寻规律，提升系统服务的准确性和可靠性。本文主要研究了以大数据驱动的交通信号灯配时排查算法，并以该算法为基础，从工程上设计并实现了一个交通大数据平台。

在交通系统中，信号灯时长的计算极其复杂，同时又对道路拥堵情况具有非常重大的影响。因此，本文先根据单车道绿灯浪费时间估计算法，利用停车线过车数据点的密度来估算绿灯浪费时间。接着介绍了单车道绿灯不足时间估计算法。在此基础上，本文继续研究了信号灯配时方案相位设计排查算法，最终通过对各个路口的大量过车数据进行分析，指出问题，并且提供相应的优化建议。

最后，利用Python实现了上述算法，并且在Django框架的基础上，利用可视化技术架构出一个交通大数据分析网站，并且稳定运行于服务器上。

**关键词：**大数据，智慧交通，可视化，网站设计

ABSTRACT

Modern intelligent transportation systems are often based on big data, constantly analyzing a large number of users and equipment data, and looking for patterns to improve the accuracy and reliability of system services. In this paper, we mainly study the algorithm of large-data-driven traffic signal timing and use this algorithm as a basis to design and implement a traffic big data platform from engineering.

In the traffic system, the calculation of signal duration is extremely complicated, and at the same time it has a very significant impact on road congestion. Therefore, in this paper, according to the single-lane green light waste time estimation algorithm, the green light waste time is estimated by using the density of the parking line data points. Then introduced the single-lane green light shortage time estimation algorithm. On this basis, this paper continues to study the phase design and troubleshooting algorithm of the signal lamp timing plan. Finally, it analyzes a large number of passing data of each intersection, points out problems, and provides corresponding optimization suggestions.

Finally, the algorithm is realized by Python, and on the basis of the Django framework, a large traffic data analysis website is constructed by visualization technology, and it runs steadily on the server.

**Key Words**: Big data, intelligent transportation, visualization, website design

目录页预留

参 考 文 献-临时版本

1. J. Zhang et al., “Data-driven intelligent transportation systems: A survey,”IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1624–1639,Dec. 2011.
2. 张昕,曾鹏,张瑞,张帆.交通大数据的特征及价值[J].软件导刊,2016,15(03):130-132.
3. 舒采焘，张孜 . 新型城市化背景下的先进交通信息服务体系构建 [J]. 交通科技与经济，2016，18（5）：21-25.
4. 关积珍 . 对北京奥运公众交通信息服务的探讨 [J]. 交通运输系统工程与信息，2008，8（6）：61-66.
5. 朱昊，王磊，张会娜 . 世博交通决策支持信息服务系统研究 [J]. 城市交通，2010，8（5）：84-88
6. 任广震,侯进,王献.MVC模式在B/S结构政务系统的应用研究[J].计算机应用与软件,2014,31(08):54-58+74.
7. 朱爱红，余冬梅，张聚礼． 基于 B/S 软件体系结构的研究［J］． 计算机工程与设计，2005，26(5):1164-1166．
8. 绪 论
   1. 课题研究目的和研究意义

现代大城市的交通拥堵是一个世界性难题，即使是在高度现代化的发达国家，城市交通的拥堵问题也没有得到根本的解决。交通问题是一个大城市经济发展的命脉，在现代都市，随着运输系统的快速发展，交通已经成为人类生活的重要组成部分并且对人们的生活质量产生了深远的影响，甚至成为了城市发展的瓶颈。调查显示，大约平均有40%的人每天至少需要花费1小时来通勤。为了改善交通状况，人们不断尝试各种方法，其中，基于大数据的智能交通系统极具前景。[1]

近年来，“大数据”一词被越来越多的提及，其中就包括交通大数据。大数据具有鲜明的“4V”（Volume、Variety、Value、Velocity）特点[2]。由于交通大数据来源、种类的多样性，使其特征更加鲜明且复杂，而且交通大数据需要面对大众，往往具有可视化的需求，一般来说，交通大数据有如下四个特点：

1) 数据量巨大，采集渠道广泛，且部分数据需要长期储存。 不同于其它系统，交通大数据系统的数据来源非常广泛，现代城市道路系统布满了各种传感器、监控探头，它们每天都会采集海量的数据，而且数据种类多样，且需要长期储存。

2) 对数据实时性和处理速度要求极高 交通大数据网站的作用之一就是实时分析城市道路的交通状况，并为用户提供出行建议和指导，因此需要快速处理数据并且实时响应。

3) 数据模态多样化 由于交通大数据来源广泛，因此其类型也极其多样化，这对数据分析系统的兼容性提出了很高的要求

4) 有可视化需求 不同于传统的报表形式，交通大数据的分析结果往往需要以可视化的形式展现给用户。

传统交通系统往往由于历史包袱的沉重，导致了其架构非常重量级，可维护性差，界面难用，交互体验差，而且由于技术的局限性，不能准确有效的分析城市道路交通系统。但如果由传统交通系统辅以大数据技术，就能够推进交通数据采集、分析、管理利用等方面的巨大进步。现代化的智慧交通系统大多采用数据可视化技术，用户可以在界面上清晰的看到实时的系统分析结果，并提供相应的建议，大幅提升了交互体验。

* 1. 课题研究现状
     1. 国外研究现状

日本从 1990 年开始交通信息服务系统（VICS，Vehicle Information and Communication System） 的研究和建设，已覆盖东京等大城市及主要高速公路。自 2002 年 VICS 中心可通过手机终端、掌上电脑、个人电脑、车载终端和电视接收器等多种途径提供交通拥堵、交通事故、道路施工、广域最优路径建议、天气状况及停车场信息等多样化的信息服务。[3]新加坡则拥有出行者信息服务系统，能够为用户提供准确实时的地铁、公交等服务信息。除了车辆的发车时间和预计通勤时间，用户还可查询到任意两地间的最少周转、最低票价或最快抵达的推荐交通路线和相应票价信息。

* + 1. 国内研究现状

我国的交通信息服务系统建设以北京、上海为典型代表。

2006 年，北京市交通委组织实施交通部公众出行信息服务系统示范工程建设，开通北京公众出行网（www.bjjt.cn），在整合交通行业信息资源的基础上，为公交乘客和自驾车出行者提供实时、动态和综合性的交通信息服务。2007 年，在示范工程基础上开展北京市公众交通信息服务系统一期工程建设，为 2008 年奥运会提供公众交通信息服务奠定基础。[4]

2006 年，上海率先建设市级交通综合信息平台，全面、实时整合和处理全市道路交通、公共交通、对外交通领域车流、客流、交通设施等基础信息资源。在此基础上，为保障 2010 年世博交通高效有序运行，建设和完善世博交通决策支持信息服务系统、世博交通网、世博交通指南、电台电视台、世博交通信息咨询服务热线、路侧可变信息标志、手机与车载导航、自助查询触摸屏等多种信息服务方式。为世博交通管理者、决策者及广大公众提供全面、实时、准确的世博交通信息服务，对世博交通安全保障起到关键性作用。[5]

上述都是国内外对智慧交通系统的早期探索。而近年来，数据可视化技术不断发展，也为交通数据分析结果提供了越来越多的展现方式。例如，基于MIT协议开源的图形库chart.js，它使用HTML5 Canvas元素构建图表，性能优越，上手简单，轻量级，无任何第三方依赖。国内厂商近年来也推出了一些可视化产品。例如，百度基于Apache2.0协议开源的图表库echarts.js，该图表库底层基于ZRender（一个轻量级的Canvas类库），提供直观，生动，可交互，可个性化定制的数据可视化图表。这些可视化技术都能方便的构建用户友好型的界面来展现交通大数据的分析结果，大大提升用户的交互体验。

* 1. 本文主要章节概述

章节概述预

第2章 理论基础

2.1 单车道绿灯浪费时间估计算法

单车道绿灯浪费时间估计算法是数据驱动信号配时排查算法的基础。其基本思想是：利用停车线过车数据点的密度来估计绿灯浪费时间。利用该算法，我们可以有效的估算单车道的绿灯浪费时间，基本算法如下：

***输入：***感兴趣时间段内，车道停车线处的过车数据；

***输出：***车道的绿灯浪费时间；

***可调参数***：绿灯时间网格化的网格长度，数据点密度阈值。

***具体步骤：***

1. 将所属相位的绿灯时间以固定长度等分，得到如图2-1所示的红色矩形（高为，宽为感兴趣的时段长度）
2. 计算每个红色矩形中的数据点个数，其中对应绿灯结束时的第一个矩形，以此类推；
3. 若，则绿灯浪费时间为

图2-1 单车道绿灯浪费时间估计算法原理

注1：算法中步骤c可以理解为：从绿灯结束时对应的矩形开始查找，如果连续i个矩形中的数据点个数小于阈值，则说明这些时间段内过车数量明显不足，所以不妨认为这i个秒的绿灯时间都是浪费的。

注2：算法中的可调参数网格长度建议选取为3秒。该经验值通过一个交叉口的小样本数据分析得到，还需要更多数据的验证。

注3：算法中的可调参数数据点密度阈值的选取需要非常仔细慎重。图2-2显示了一个交叉口所有车道的网格化绿灯时间通过的车辆数(选取为3秒)。该交叉口为传统四相位配时。由图2-2可以看到如下结论：

1. 不同车道的通行能力不同(东西向直行车道的车辆数明显多于其他车道)，所以需要对不同车道选取不同的，而且的值应当由过车数据来决定。由于是一个经验值，因此需要对大量过车数据进行分析来得到。经过大量实验模拟，得出一种可行的选取为：



即选取绿灯开始时矩形中的车辆数乘以0.4作为阈值。系数0.4是为了使得绿灯浪费时间的判断更为严格。图2-2中，所有车道的为横坐标为2的点。

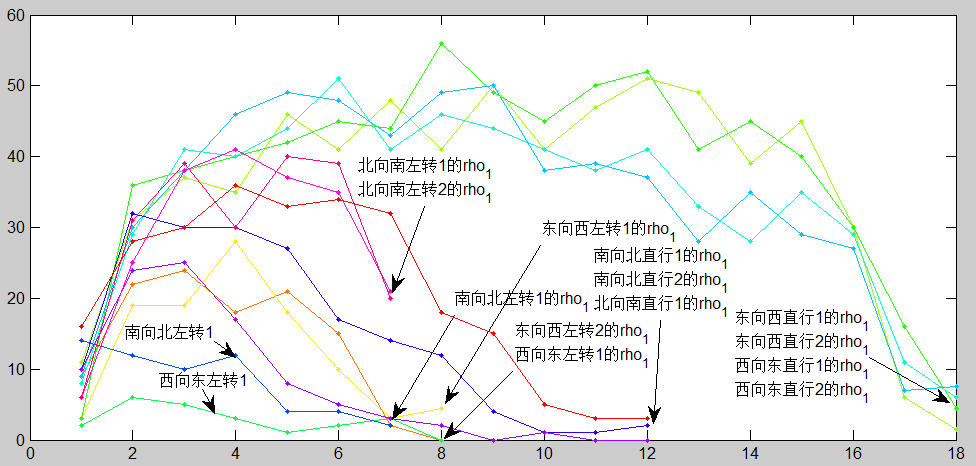
1. 对于过车数量非常少的车道，即平均每个周期通过的车辆数少于3辆(如南向北左转1和西向东左转1)，上述算法并不适用，因为该种情况下的完全不能体现排队消散时的流率。此时，可以直接利用绿灯时间减去最小绿灯时间作为绿灯浪费时间。

图2-2 某交叉口所有车道的

得到车道的绿灯浪费时间后，就可以利用其判断该车道的绿灯时间是否合理，判断依据如下：

* 若绿灯浪费时间等于，则合理；
* 若绿灯浪费时间等于0，则不合理，建议增加绿灯时间；
* 若绿灯浪费时间大于，则不合理，建议减少绿灯时间；

2.2 单车道绿灯不足时间估计算法

理想状况下过车/旅行时间预测（绿灯时间增加）

***假设 1：*** 理想情况下车辆的驾驶行为满足如下假设：所有车辆行驶速度相同。因此，所有车辆在不遇到红灯时，旅行时间都为，称为*标称旅行时间*。

图2-3显示了理想情况下绿灯时间增加引起的车辆行驶过程变化。图中，红色线段表示原来的红灯时间，桔色线段表示绿灯增加后的红灯时间，14条黑色实线轨迹表示在原来配时下14辆车辆的时空轨迹，14条蓝色虚线轨迹表示在绿灯增加后14辆车辆的时空轨迹。图底部的标明了车辆时空轨迹的起点；图顶部的数字标明了车辆时空轨迹的终点，括号外的数字标明了绿灯增加后车辆时空轨迹的终点，括号中的数据标明了原来配时下车辆时空轨迹的终点。

图2-3 理想情况下绿灯时间增加引起的车辆行驶过程变化

通过对图2-3的分析可知，当绿灯时间增加时，新的过车时间和新的旅行时间的变化可以分为如下三类：

1. 若前面有n辆车提前通过，则第辆车的新过车时间和新旅行时间可以用下面的方法计算：

 （1）

其中和为第辆车在原配时下的过车时间和旅行时间。例如：

图2-3中的v1：；车辆v2：。

1. 若由（1）式得到的新旅行时间小于标称旅行时间，则说明提前通过的车辆较多，故更新提前通过的车辆数，再由（1）式重新计算和。
2. 若由（1）式得到的新时空轨迹可以在增加的绿灯时间通过，则说明提前通过的车辆数过少，故更新提前通过的车辆数，再由（1）式重新计算和。

结合以上结论，在理想情况下（假设1满足的情况下），若绿灯时间增加，则过车时间/旅行时间的预测算法如下：

|  |  |
| --- | --- |
| function [,] = estPtTt() | |
| 输入: | 某车道的过车时间数据, 为数据点个数  某车道的旅行时间数据, 为数据点个数  获得数据时的周期  获得数据时的绿灯时间, 为绿灯开始时刻，为绿灯结束时刻  增加的绿灯时间  标称旅行时间 |
| 输出: | 新配时方案下，过车时间的预测值  新配时方案下，旅行时间的预测值 |
| % 过车时间提前的车辆数  % 上一辆车的轨迹是否是新产生的  for  % 更新过车时间  % 更新旅行时间  while  and  % 旅行时间必须大于等于标称旅行时间    % 更新过车时间  % 更新旅行时间  end    % 判断更新后的过车时间能否在增加的绿灯时间内通过  if      else      end    if  and  and          else    end  end | |

表2-1 过车时间/旅行时间预测算法(绿灯时间增加，周期不变)

2.3 配时方案相位设计排查

第3章 网站基础架构

3.1 传统C/S体系架构

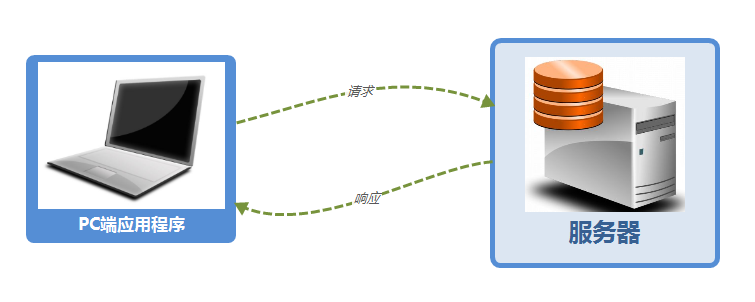
传统的C/S体系架构即客户端/服务端架构，如图3-1所示。这种体系架构

图3-1 传统C/S架构体系

在20世纪80年代由美国的Borland公司设计开发[6]，结构清晰简单。其中，PC端应用程序负责展示用户界面，完成交互逻辑，并向服务器端发送相应的请求。服务器端则负责业务逻辑的处理与数据的持久化，并向客户端返回对应的结果。

这种架构的好处和坏处都显而易见，好处有：应用程序直接访问服务器，可以自定义网络协议，运行速度快，服务器端负载小。不足之处在于：这种架构跨平台能力差，需要为不同的终端专门编写不同的客户端程序；扩展性低，维护困难。此时，所有应用程序，数据库，文件等资源全部在同一台服务器上，系统耦合较为严重。随着现代互联网技术的不断发展，这种架构逐渐无法满足用户的需求，于是，出现了B/S架构。

3.2 B/S三层架构

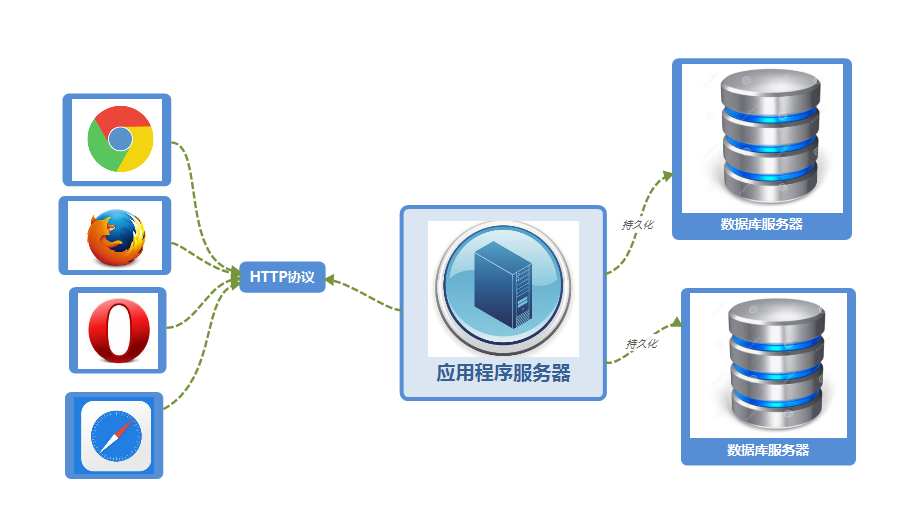
B/S架构体系[6]即浏览器/服务器体系结构，最早由美国微软公司(Microsoft)设计研发。不同于传统的C/S架构，该架构不依赖于专门编写的客户端应用程序或者APP，而是将大部分终端设备都有的网页浏览器作为用户界面的展示窗口。这是一种具有划时代意义的系统架构，B/S架构的出现，大大减少了客户端的维护与升级成本，并且在大局意义上统一了客户端的表现结构（浏览器），为后来的互联网时代奠定了基础。B/S架构的基本原理如图3-2所示。

图3-2 B/S架构基本原理图

图3-2中，第一层是用户浏览器层，主要负责展示界面和交互逻辑。中间层为应用程序服务器，专门负责处理业务逻辑，浏览器通过HTTP协议与浏览器进行通信。以Java为例，常见的应用程序服务器有Tomcat、Jboss等。需要注意的是，应用程服务器专门用于处理业务逻辑，而不负责数据持久化，这样做的好处是可以专门为服务器进行参数调优和硬件适配。底层就是数据库服务器，依据用户规模的大小可以做成分布式，由专门的数据库管理员(DBA)进行管理，负责用户数据的存储和维护。由于分层体系架构的原因，数据库服务器也可以专门进行优化，同时也更方便进行数据备份，主从复制，读写分离，异地热备等工作。

基于B/S架构的应用，有效弥补了C/S架构的不足，能够让开发人员更方便的开发出交互性强、跨平台的应用，同时也大大降低了日后的维护和扩展成本。

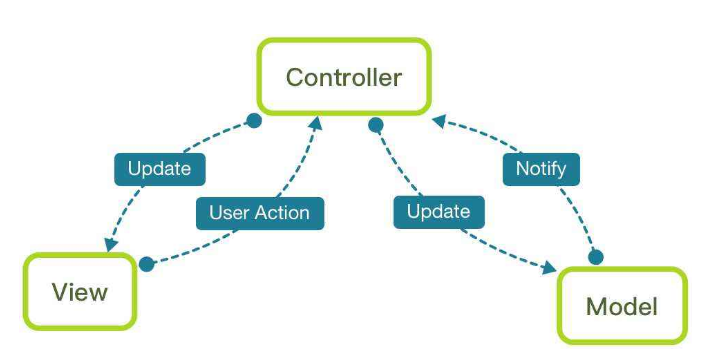
3.3 MVC设计模式

3.3.1 MVC模式简介

MVC的概念是在20世纪70年代由Xerox PARC的Trygve Reenskaug[8]提出。

MVC即“Model-View-Controller”的缩写，中文一般称为“模型-视图-控制器”。

每个部分分别负责不同的功能，其基本原理如图3-3所示。

图3-3 MVC模式基本原理

3.3.2 MVC基本原理

模型（Model）是整个MVC模式的核心，在一个设计合理的软件架构体系中，模型层中应当封装了系统的业务逻辑（关于用户数据持久化的部分，将在本章的下一节介绍）。在经典的MVC模式中，事件由控制器（Controller）处理。每当服务器接收到HTTP请求，控制器将会根据请求路径与请求方法，交给对应的Controller来处理，当Model处理完业务之后，再将结果转发给对应的视图。

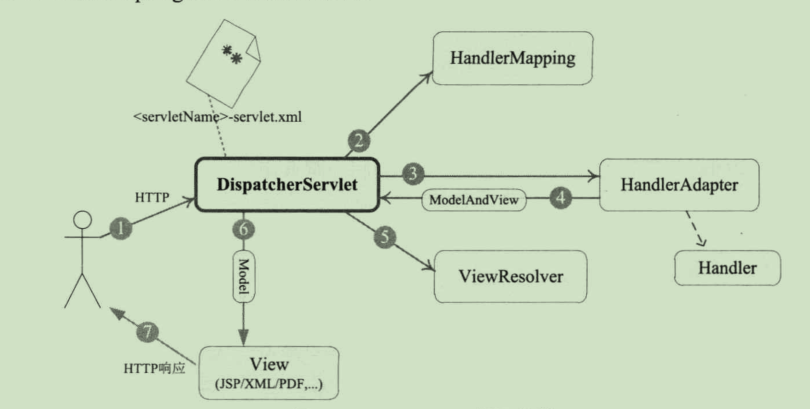
以SpringMVC为例，从接收到请求开始，框架内各个模块组件通力合作，有条不紊的完成各自的工作，直到返回对应的结果，如图3-4所示。

图3-4 SpringMVC框架模型

DispatcherServlet是整个MVC框架的核心，它负责整个框架的各个组件之间的协同配合，下面将详细分析整个流程：

1. 当一个客户端（浏览器）发出HTTP请求，应用程序服务器（通常是指Tomcat）接收到该请求，如果该请求匹配相应的规则，则由服务器将该请求转交给DispatcherServlet处理。这相当于MVC框架的Controller。
2. 当DispatcherServlet接受到该请求后，会根据请求的请求信息（请求路径、请求方法、请求参数、Cookie等），来匹配对应的请求处理器（Handler）。其实这就相当于一个Model。
3. 当请求处理器完成了业务逻辑之后，将返回结果给DispatcherServlet，返回结果是一个包含了视图（View）信息，并根据对应的数据进行视图渲染。

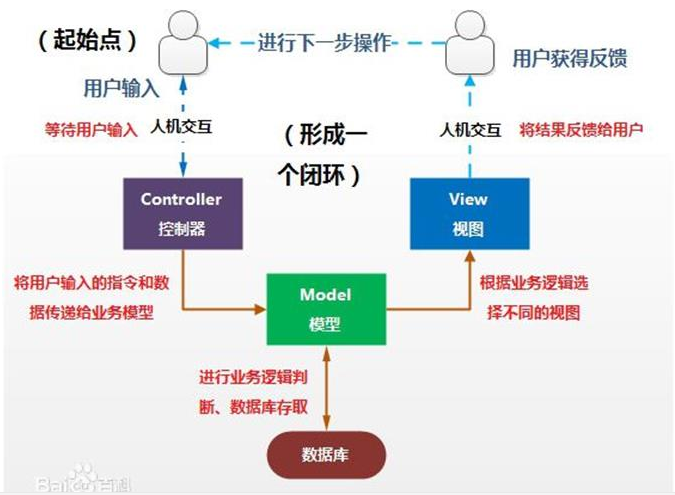
由SpringMVC的工作流程可以看出，HTTP请求不会直接交给model，而是先同意将请求交给Controller，再由Controller根据预先定义的规则转发给Model，如果此时业务逻辑的需求发生变化，我们只需要将Model重新实现即可。本质上来看，Controller的作用就是用来协调Model和View。最终，结合B/S三层架构，可以得到如图3-5所示的架构模型：

图3-5 MVC模式闭环模型

3.3.3 MVC的优势与不足

从上一小节对SpringMVC运行流程的分析者我们可以发现，基于MVC模式的优势有：

* 可复用性高，易维护易扩展：多个View可以对应同一个Model，可以有效增加代码的可复用性，减少代码的冗余。同时，一旦需求发生改变，独立于其它模块的Model也更易于维护和扩展。
* 耦合性低：我们可以发现，Model返回的数据与View层的显示逻辑分离，任一模块的改动都不需要修改别的模块的代码，例如当展示界面的需求发生变化时，我们仅需要修改View层的代码即可。更重要的是，由于数据与显示逻辑的分离，我们可以根据自己的需求采用第三方的显示技术，例如，Java界常用的Freemarker、Velocity和Themeleaf模板引擎。
* 更符合软件工程的精神：不同的层相互独立，各司其职，每一层之间又具有极高的内聚性，这使得开发和维护接口的工作量大大降低，能够使后端开发者更加专注于业务逻辑，前端开发者更加集中于表现形式的开发上。同时也有利于通过工程化和工具化的方式来管理代码。

与此同时，虽然MVC模式有这么多好处，但也带来了一些负面的问题，其缺点主要是：

* 完全理解MVC的设计模式比较复杂：我们并不能保证所有开发者都具有良好的实践经验。使用MVC模式需要大量进行的设计与计划，这才一定程度提高了准入门槛。
* 不适合中小型的应用程序：对于简单的应用程序，如果强行使用MVC设计模式，使得各个层之间分离解耦，反倒增加了结构的复杂性，使得测试困难，增加过多的冗余代码，降低系统运行效率。