

基于 MVC 架构的电工电子网络实验室的 B/S 实现*

张 欢, 朱善安*

(浙江大学 电气工程学院, 浙江 杭州 310027)

摘要:为了解决传统实验教学中仪器设备不足、实验场地有限、学生不能及时进行实验等问题,以远程网络实验室的形式,提出了基于 B/S 的网络实验室构架,为用户提供了通过互联网从浏览器登录、进行远程实验的平台。详细介绍了构架前端以 Adobe Flex 技术,在 MVC 架构下实现 Browser 与 Server 的网络通讯,并从客户端抽象出了中间层(数据交换层),完成了整个系统的数据收发工作,服务器则负责转发从客户端到实验端的数据包。研究表明,该平台具有良好的可扩展性、可维护性以及良好的二次开发接口,可以根据需要扩展新的实验。

关键词:B/S; 远程实验平台; 电工电子网络实验室; MVC; Flex; 数据交换层

中图分类号:TM1; G642; TP303

文献标志码:A

文章编号:1001-4551(2013)09-1159-05

B/S implementation of internet-based electrical engineering lab based on MVC architecture

ZHANG Huan, ZHU Shan-an

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: Aiming at solving the problems of traditional experimental teaching in insufficient equipment resources and limited experiment room, the Browser/Server(B/S) mode of internet-based electrical engineering lab(iEELab) was proposed in distance education with the form of remote network laboratory. The remote experiment platform was provided for users to login through various browsers and make remote experiments freely. Front-end, using Adobe Flex technique and managing network communication from browser to server based on MVC architecture was elaborated. A middle layer called data exchange layer was extracted from the front-end, which took responsibility for data sending and receiving, while the server only managed to transmit data package. The results indicate that the whole system has good expansibility, maintainability and secondary development interface. New experiment modules can easily be added to the lab.

Key words: B/S; remote laboratory; iEELab; MVC; Flex; data exchange layer

0 引 言

随着互联网技术的飞速发展,网络远程教育越来越受到人们的关注。而在传统实验环境中,由于存在动手实验室^[1]场地设备费用高昂、使用人数有限等问题,导致学生不能及时进行实验。为了弥补这一缺点,远程网络实验室应运而生,这些网络实验室主要分为

两大类:一类是基于真实物理设备的远程实验室,让学生通过网络共享实验资源;另一类是虚拟仿真实验室,可进一步缓解实验室设备资源的压力^[2-3]。

远程实验室基于真实的物理设备和网络环境,用户可以通过 Internet 远程访问实验服务器,进行实验操作和研究。用户可以在任何时间、任何地点,通过计算机进行远程实验操作,共享实验资源,这大大缓解了

传统物理实验室资源紧张的局面,有效提高了实验资源的利用率。

在 Internet 的流行趋势下,以往的 PC 终端、C/S 都无法满足当前的全球网络开放、互连及信息共享的新要求,于是就出现了 Browser/Server,浏览器/服务器模式(B/S)。B/S 最大的优点就是所有客户端只需装上操作系统、网络协议软件、浏览器即可,使客户端不再受平台的约束^[4]。因此,浙江大学网络实验室致力于研究开发 B/S 模式下的网络实验室是顺应时代发展要求的。这样可以提高实验效率,降低实验成本,使实验室的仪器设备的使用效率最大化^[5]。

本研究重点从客户端角度讨论如何利用 Adobe Flex 技术,在 MVC 架构下设计出客户端界面以及客户端与服务器、控制端的数据交换,并说明采用 MVC 的优势。

1 系统设计

浙江大学网络实验室采用了流行的双 C/S 架构,包括客户端、服务器端和控制端。以前,用户只能采用客户端软件方式登录网络实验室,为了使用户也能使用浏览器登录,本研究在原来系统的基础上增加了 B/S 架构,以便更多用户能进行远程实验。

1.1 系统结构

浙江大学网络实验室的拓扑结构如图 1 所示。它主要包括以下 3 个部分:

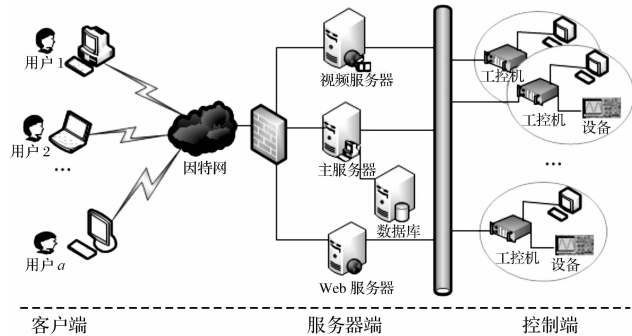


图 1 网络实验室的物理拓扑结构

(1)客户端。有两种模式,用户使用客户端软件登录,这就是已经实现的 C/S 模式;通过浏览器登录实验系统,即正在开发的基于 B/S 模式的前端系统。前者基于 MFC 框架开发;后者采用 Adobe Flex 技术开发,每门实验都是一个单独的 Flex 插件,嵌入到统一的管理框架,任何子系统的增加与删除都不影响其他实验的正常运行,各个实验模块之间互相独立、互不干扰,这样的实现便于扩展开发和系统优化与维护。

(2)服务器端。为了提高系统的稳定性和可靠

性,网络实验室采用了多服务器架构来提供不同的 Web 服务。

①主服务器。即应用程序服务器,负责监听用户的 Socket 连接并响应用户请求,主要完成用户登录、实验注册和数据转发等工作,是整个服务器系统的核心。

②Web 服务器。使用开源的 Apache Tomcat 作为 web 服务器容器,部署网络实验室相关的资源。

③数据库服务器。使用 MySQL 关系型数据库来存储管理用户和实验信息,通过 Hibernate 方便地操纵数据库,完成数据持久化的重任。

④视频服务器。使用 AXIS 2400 视频服务器,为远程实验提供可视化服务,使用户有亲临现场的感受,如图 2 所示(左下角的视频窗口)。

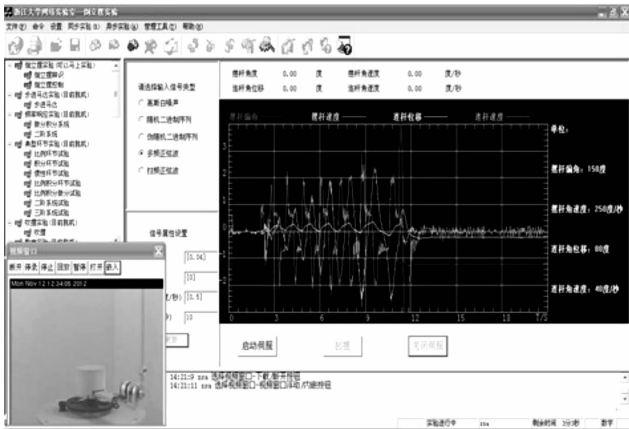


图 2 控制类倒立摆实验

(3)控制端。即实验端,实物远程实验平台,网络实验室现有 23 组实物平台,每组平台包括工控机和实验设备两部分。工控机主要运行 2 个进程:管理进程和监控进程。管理进程用来接收指令和传送实时数据;监控进程用来监控设备状态和网络状态,是实现网络实验室 24 h 全天候开放的重要保障。

1.2 工作流程

用户以 B/S 方式进行实验的典型流程如图 3 所示。首先,用户从网站通过浏览器以用户名和密码登录,不同的用户拥有不同的权限;用户在页面实验列表中选择需要做的实验,在实验面板中选择使用已有的器件搭建新的实验;开始实验后,实验插件会将用户的请求以及实验的相关数据封装起来通过系统的数据交换层发往主服务器;最后,主服务器将请求转发给对应的实验设备控制模块,控制模块负责接收指令和传送实时数据。实验设备响应之后,将实验结果数据打包,按相反的数据流方向将数据传回实验插件进行结果的可视化显示。

数据交换层是用 JavaScript 实现的一个数据传输

中间层,是整个系统的通讯枢纽,它是从前端 Flex 插件抽象出来的一个统一接口,每门实验的插件都可以调用它完成数据发送和接收,还可以为登录实验、登记实验、开始实验、保存场景、清空场景等通用功能提供数据交换支持。整个系统的设计高内聚低耦合。这种系统符合模块化开发的要求,使得系统维护、扩展以及单元测试都变得容易。

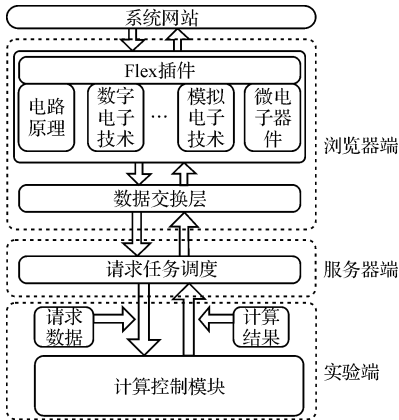


图3 系统工作流程

2 在 MVC 构架下的开发与实现

早期微软所推出的 MFC Document/View 架构是对于 MVC 模式的实现,MFC 将程序分成 CView 以及 CDocument 两大类型,其中的 CDocument 对应 MVC 中的 Model,CView 相当于 MVC 中的 View + Controller,再加上 CWinApp 类型,构成 MVC 3 大项。MVC 的运用很广,目前流行的开源框架 Struts^[6-8]是 MVC 中的一个经典产品,它是基于 Java EE Web 应用在 MVC 框架下的具体实现。鉴于 MVC 的上述优点,本研究也选择 MVC 框架来开发网页应用程序。

MVC(Model-View-Controller)架构采用了“分治”的思想^[9],将数据模型、数据显示以及数据控制逻辑进行了分离。MVC 是一种 3 层架构模式,把软件系统分成 3 个基本部分:模型(Model),视图(View),控制器(Controller)。从结构上看,MVC 被分成 3 层;从功能上看,三者构成了两个控制回路,基本 MVC 结构图如图 4 所示。

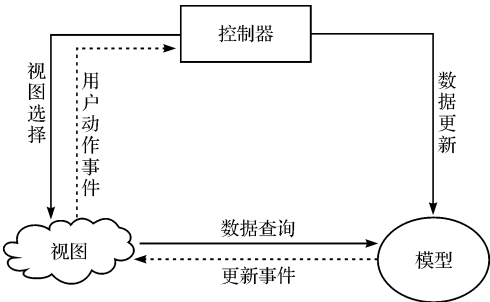


图4 基本 MVC 结构图

2.1 数据层 (Model)

“数据模型”(Model)用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。“模型”不依赖“视图”和“控制器”,即模型不关心它将被如何显示或是如何被操作。但模型中的数据一旦变化会通过一种刷新机制被呈现。为了实现这种机制,那些用于监视模型的视图(监听对象)必须事先在模型上注册,视图从而可以知晓在数据模型上发生的变化。

在 Model 的实现中,关键在于监听对象和数据模型之间的注册。在 Flex 里,有很多系统事件和自定义事件,如连线、拖拽等,都可以使用 addEventListener 方法来注册。在该系统中,主要有 2 个监听对象:OperateCanvas(实验操作面板)和 ApparatusSvg(器件 SVG 矢量图)。研究人员期望所有的监听器都能统一管理,因此本研究采用适配器(Adapter)来实现。因此本研究定义了一个 ListenAdapter 类,列举 2 个主要的静态方法如下:

```
static public function loginDragApparatus ( container: OperaleCanvas ):void {
    container.addEventListener( DragEvent. DRAG_DROP, container. dragDropHandler );
    container.addEventListener ( DragEvent. DRAG_ENTER, container. dragEnterHandler );
    container.addEventListener ( DragEvent. DRAG_OVER, container. dragOverHandler );
}

static public function loginApparatusDrag ( container: ApparatusSvg ):void {
    dontainer.addEventListener ( MouseEvent. CLICK, container. showResult );
    container.addEventListener ( MouseEvent. MOUSE_DOWN, container. dragIt );
}
```

2.2 视图层 (View)

视图层能够实现数据有目的的显示,具有和外界交互的功能,主管应用系统与外界的交互。在视图中一般没有程序上的逻辑,为了实现视图的刷新功能,视图需要访问它监听的数据模型,因此应该事先在被它监听的数据那里注册。

在视图的实现中,有 3 个关键点。首先是在视图的初始化函数中,把视图注册到指定的模型中去(如 ListenAdapter. loginDragApparatus(this));其次,写一个针对监听对象的刷新函数(如 public function dragEnterHandler(event: DragEvent): void);最后,为视图写一个根据传入对象的类型来显示的方法 showResult。通过上

述 3 个要点,视图实现了发送用户动作到控制器,在 Controller 的作用下接收数据的更新并刷新显示。

2.3 控制层 (Controller)

控制层起到了不同层面间的组织作用,用于控制应用程序的流程。它处理事件并作出响应。“事件”包括数据模型的改变和用户行为。MVC 响应机制封装在 Controller 对象中,控制器负责转发请求并对请求进行处理。

对于 Flex 来说,控制器捕获用户的操作请求,将其转换成相应的应用程序动作(如将用户进行鼠标单击这一动作转换成 Flex 语言中的 MouseEvent.CLICK 事件类型,鼠标按下转换成 MouseEvent.MOUSE_DOWN,鼠标弹起转换成 MouseEvent.MOUSE_UP 等),然后通过事件分发器 EventDispatcher 将事件调度到事件流中,当流经对象为目标对象时,系统就调用监听函数进行处理。

2.4 用 Flex 实现的 MVC 流程

由于 Flex 是事件驱动的,本研究利用其事件流体系,首先在代码中针对不同的事件和视图建立事件侦听器,以使侦听器能够接收事件通知,当事件被触发时,就进入事件触发时的回调函数完成特定功能,依靠 Event Bus 管理整个事件流,通常都是利用 Flex 自身的冒泡事件体系层层传递事件来管理的。

MVC 的基本流程如图 5 所示。

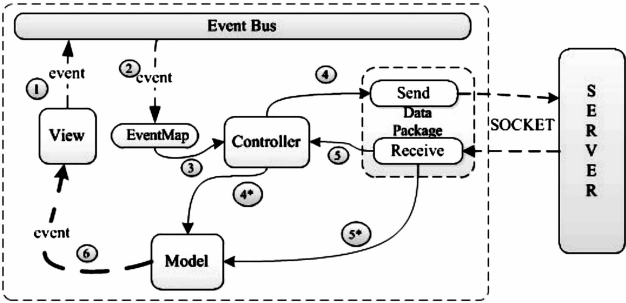


图 5 MVC 基本流程

当用户进行界面操作时,通过查找 EventMap 的对应关系,实验系统根据触发的事件类型找到对应的侦听器函数,在函数中完成模型数据的改变和业务逻辑的处理。模型将改变通知它的视图,而视图将与模型通信以访问这些更新的数据值。在设计中笔者将 Model、View 的对象分离,使一个对象的改变能影响另一个对象,而这个对象并不需要知道那些被影响的对象的细节,这种设计被描述成观察者 (Observer) 模式。而 View-Controller 关系是策略 (Strategy) 模式的一个例子,一个策略是一个表述算法的对象,如果需要静态或

动态地替换一个算法,这种模式是非常有用的。与服务 器交互时,由抽象出来的数据交换层来完成与服务 器的数据包的发送和接收。服务器返回的结果可以直接传递给 Model 或者通过 Controller 对象间接传递给 Model。

3 系统实例

整个 B/S 系统的登录界面如图 6 所示,用户通过任意浏览器就可登录网络实验室进行实验操作。



图 6 系统登录界面

用户自行设计的实验并在实验操作面板中搭建的电路图如图 7 所示;右下角是网络实验室服务程序主控面板,显示说明一个名为“zsa”的用户开始进行数电半加器实验。

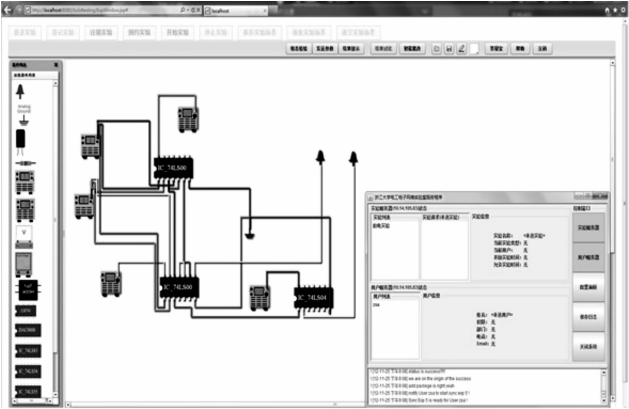


图 7 数电类半加器电路搭建

在经过实验电路的组态检验后,提示电路没有错误便可开始发送实验数据,用户点击“参数发送”按钮,等待实验端响应后,实验数据返回,靠左的二极管变亮,表示返回数字信号“1”。说明在输入“1”和“0”后,“和输出”返回“1”,“进位输出”返回“0”。用户点击控制条中的“结果对比”按钮,会弹出对应的逻辑原理图,跟 B/S 实验结果对比,这样就可以得出自己设计的电路与原理图之间的差别,半加器实验结果显示如图 8 所示。

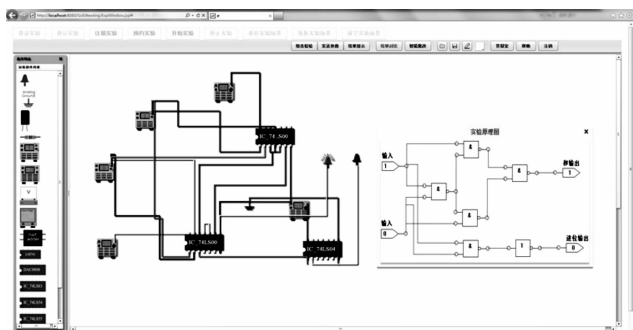


图8 半加器实验结果显示

该实例表明,采用 MVC 构架来开发网络实验室系统,是理想的选择。通过 3 层解耦可以很方便地处理整个业务逻辑,例如结果返回时,客户端从控制端返回的数据流中解析出操控二极管行为的信息,然后触发 OperateCanvas 上对应的二极管 SVG 矢量图由灭(default 状态)变为亮。

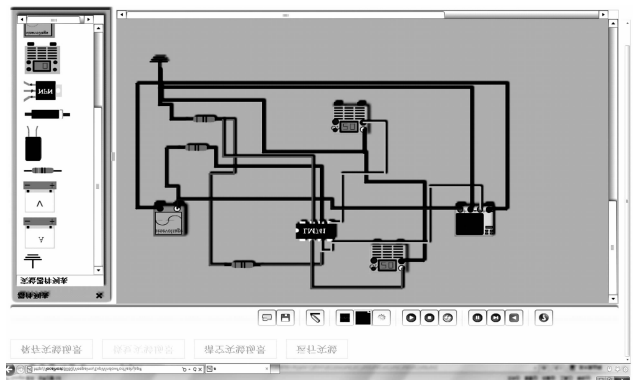


图9 模电类同相比例运算放大电路搭建

模拟电子技术实验中的同相比例运算放大电路的电路搭建如图 9 所示。输入是正弦波,输出是经过一定比例放大后的正弦波。输入与输出的波形及其幅值如图 10 所示,图中可以看出输入幅值为 0.5 V,输出幅值为 1 V,放大倍数为 2。

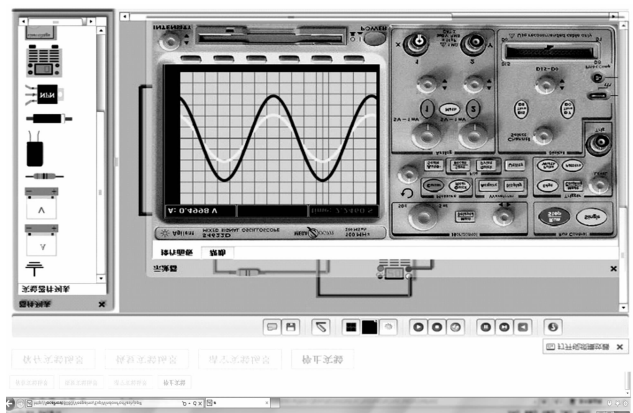


图10 示波器结果输出

4 结束语

本研究重点从客户端角度讲述了在 MVC 构架下 B/S 的设计与 3 层解耦实现。通过使用这种设计方式,将避免那些可能导致使底层代码变得繁杂混乱的大量属性文件和辅助类。

浙江大学已经建成一个基于物理设备的网络实验室 NetLab^[10],NetLab 覆盖了电工电子类的 6 门课程,并从 2005 年开始向浙江大学远程教育电气工程自动化专业的学生开放。每年分布在全国各地的 124 个远程学习中心近千人通过 NetLab 共享浙江大学实验室资源。但为了更好地适应时代的发展需求,让更多用户能使用该套资源,本研究设计与整套系统兼容的 B/S 模式并在 MVC 架构下实现了网络通讯。

通过使用 B/S 方式进行实验,用户无需专门下载客户端软件,系统维护和扩展开发都极为方便。在不久的将来,网络实验室还将完善一些创新性的功能来增强用户体验,如智能批改和实现网上辅助教学、智能化管理的功能。

参考文献(References):

- [1] 曹晓欢,朱善安.面向电气工程高等教育的综合性远程实验室[J].实验室研究与探索,2010(10):348-351.
- [2] 易小琳,王鑫,丁蕾,等.网上计算机系统虚拟实验室的研究[J].计算机工程,2002,28(11):243-245.
- [3] 刘民岷,杨平,吴浩文.基于虚拟仪器的实验室建设[J].实验技术与管理,2002,19(1):93-96.
- [4] 童德利,田娟,谢琪,等.基于 B/S 模式的构件式酒店管理信息系统的设计与实现[J].计算机应用研究,2003(4):126-130.
- [5] DUAN Bing, HABIB M M. An architecture for online laboratory E-learning system[J]. Journal of Distance Education Technologies, 2006, 4(2): 87-101.
- [6] 张海玉,刘晓霞.用 Struts 开发基于 MVC 的 Web 应用[J].微机发展,2005(7):41-43.
- [7] 寇毅,吴力文.基于 MVC 设计模式的 Struts 框架的应用方法[J].计算机应用,2003,23(11):91-94.
- [8] 武宝珠,梁声灼,牛德雄.基于 Struts2 + Spring + Hibernate 架构构建 Web 应用系统[J].计算机与现代化,2009(8):43-46.
- [9] 刘克. MVC 架构及其在 Web 应用开发中的应用[J].计算机应用与软件,2006,23(7):57-60.
- [10] ZHANG Shu, ZHU Shan-an, LIU Qun. NETLAB-An Internet based laboratory for electrical engineering education[J]. Journal of Zhejiang University: Science, 2005, 6(5):393-398.

[编辑:李辉]