基于轻量级数据操作架构的应用系统①

李 皎[1,2], 梁工谦[2], 李湘眷[1]

[1](西安石油大学 计算机学院, 西安 710065)
[2](西北工业大学 管理学院, 西安 710072)

摘 要: 为了满足用户对 Web 应用程序的交互性强、响应速度快的体验需求, 使用 Ajax 技术设计实现基于 UI 组件的数据操作模式, 给出了一种基于轻量级数据操作架构的 Web 应用系统解决方案. 将界面设计、业务逻辑处理及数据访问交互相分离, 增强了程序设计的独立性, 采用 Ajax 异步通信方式可降低服务器负荷, 提高数据交互响应速度, 采用分层方式设计数据操作界面, 操作清晰, 简洁方便. 在实际应用中该架构易于现有主流 Web UI 相结合, 具有一定通用性. 文中最后采用 ExtJs 和 ASP.NET 设计实现了石化腐蚀评价系统应用实例.

关键词: 应用系统; Ajax; 轻量级数据操作架构; ExtJs

Application System Based on Lightweight Data Operation Architecture

LI Jiao[1,2], LIANG Gong-Qian[2], LI Xiang-Juan[1]

[1](School of Computer Science, Xi'an Shiyou University, Xi'an 710065, China)
[2](School of Management, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: In order to satisfy customers' experience needs with strong interactivity, fast response speed of Web applications, this paper designs and implements a data operation method based on UI components using Ajax technology, and offers a solution of Web application system based on lightweight data operation architecture. It separates interface design, business logic processing and data access to increase the independence of program design, and introduces Ajax asynchronous communication mode to reduce the load of servers and improve time response of data interaction. Using hierarchical design data interface, operation is clear, concise and convenient. In the practical application, the architecture is easily combined with existing mainstream Web UI and has some certain commonality. Finally using ExtJs and ASP.NET, application examples of petrochemical corrosion evaluation system was designed and implemented.

Key words: application system; Ajax; lightweight data operation architecture; ExtJs

Web2.0 时代的到来给 Internet 带来了空前发展, 前端用户体验显得越来越重要. 用户对 Web 应用程序的交互性和响应速度提出更高的需求. 传统的 Web 应用程序采用请求响应方式刷新 Web 页面, 即使页面只更新极少的一部分, 也需要全部刷新页面, 从而导致服务器负担重, 页面响应速度低. 传统的 B/S 设计方法已经不能满足交互性强、响应速度快的用户体验需求. 国内外对 Web 应用程序框架研究较多, 取得了一些研究成果, 它们大多是针对具体应用背景或特定开发环境. 文[1]结合具有异步传输特征 Ajax, 提出了具备动态性和松耦合特征的 Web 计算框架. 文[2]提出了一种基于 Ajax 的 Web 应用构件组装技术. 文[3-5]基于 ExtJs 研究了不同的 Web 应用程序框架. 文[6]设计实现了一种管理信息系统通用数据操作模板. 本文采用 Ajax 技术给出了一种 Web 应用系统的数据操作解决方案, 可创建 RIA 富客户端浏览、增加、修改、删除界面, 并且可和现在主流的 Web UI 结合使用, 具有一定的通用性. 该方案将界面设计、业务逻辑处理及数据访问相分离, 提高了程序设计的独立性, 降低了程序开发的难度和复杂度; 同时, 在服务器与客户端之间

① 基金项目:国家自然科学基金(41301480);陕西省科技厅项目(2015GY026, 2015GY102);陕西省教育厅项目(14JK1573,15JK1571,15JK1586)
收稿时间:2015-09-06; 收到修改稿时间:2015-10-19

采用异步通信方式，只对需要更新的部分进行刷新，避免了因整个页面提交而造成浏览器假死现象，而且不影响用户进行与数据更新无关的操作.

1　轻量级数据操作架构的应用系统设计

1.1 设计方案

图 1 给出了一种轻量级数据操作架构的应用系统设计方案，采用三层结构，从上至下分别为：界面显示层、业务逻辑处理层、数据访问层[7,8]. 界面显示层完成与用户界面操作相关的功能，界面显示层用到的组件包括数据显示组件、操作工具栏、表单及字段组件、提示信息组件. 业务逻辑处理层主要完成数据操作业务逻辑，当界面显示层的浏览界面需要显示数据时，该层向数据访问层发出 Ajax 请求，数据访问层的通用数据处理单元接收到 Ajax 请求，完成数据查询操作，并通过数据集转换单元将数据集转换成界面显示层能解析的 Json 字符串. 增加、修改、删除操作是一种与用户进行交互的操作，在增加、修改界面通常需要对录入信息进行验证，包括主键重复验证、唯一性验证、录入范围验证、录入格式验证等. 删除操作有时候是不可恢复的，具有一定的危险性，在删除前需要用户再次确认其操作. 因此在增加、修改、删除界面层和业务逻辑处理层进行交互时，加入了一个交互界面. 交互界面主要完成对录入非法信息进行提示和危险操作的提示. 当业务逻辑处理层发出增加、修改和删除 Ajax 请求时，数据访问层的通用数据处理单元接收到 Ajax 请求后，完成相应的操作. 下面结合该设计方案给出浏览、增加、修改、删除操作的设计流程.

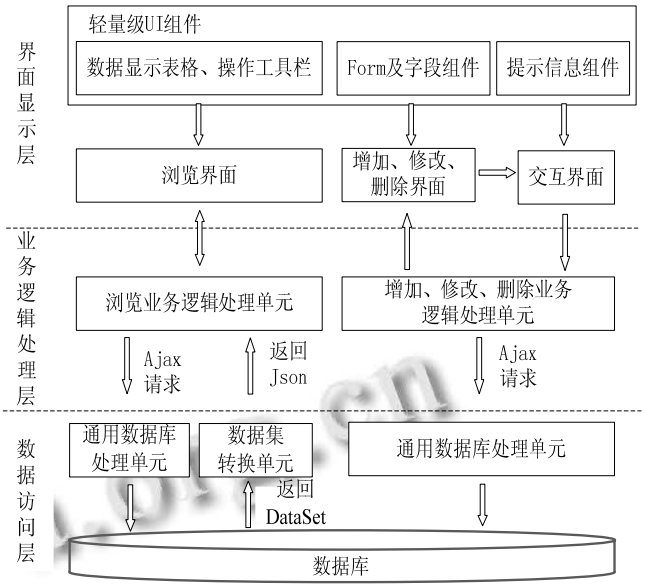

图 1　一种轻量级数据操作架构的应用系统设计方案

1.2 浏览界面设计流程

数据操作界面采用分层方式，首先呈现给用户的是数据浏览界面，数据增加、修改和删除是基于浏览界面的. 图 2 给出了数据浏览界面的设计流程. 设计流程如下：(1)浏览界面上显示的数据项需要在数据模型中定义，因此需要根据数据库表字段首先创建数据模型；(2)定义数据源，包括请求服务器的 url、请求方式、请求数据的读取格式等；(3)由于增加、修改、删除操作是基于浏览界面的，因此要定义操作工具栏；(4)根据用户的显示需求选择合适组件显示数据；(5)通过 Ajax 异步请求数据；(6)解析 Ajax 返回的 Json 字符串，将数据呈现在浏览界面中.

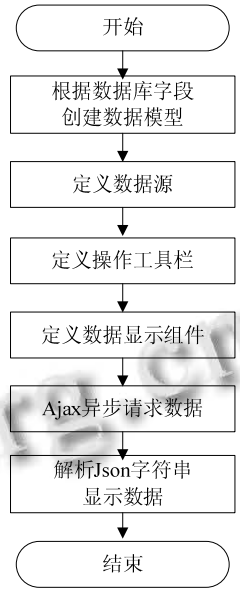

图 2　浏览界面设计流程

1.3 增加界面设计流程

在浏览界面的中点击“增加”按钮可弹出增加界面. 增加界面需要根据数据库字段的类型选用合适的组件创建录入项，并且需要对录入项进行验证，以保证录入信息的合法性. 当验证通过后，可通过 Ajax 请求保存录入项. 图 3 给出了增加界面的设计流程，图 4 给出了“保存”事件的处理流程.

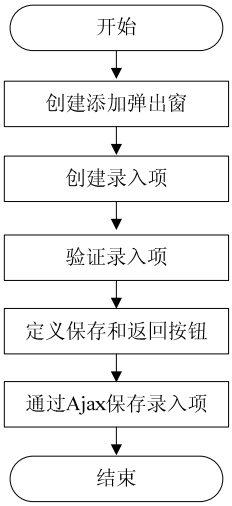

图 3 增加界面设计流程

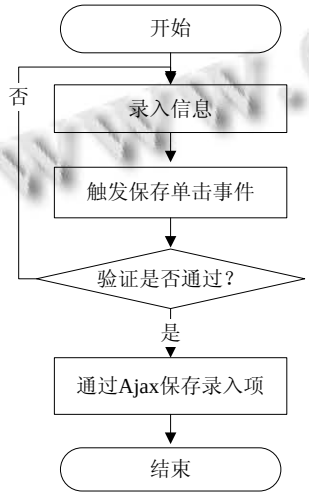

图 4 增加界面“保存”事件处理流程

1.4 修改界面设计流程

在浏览界面选中要修改的记录, 然后点击“修改”按钮可弹出修改界面. 在修改界面中, 需要呈现待修改记录的数据项, 这样可方便用户进行修改操作. 修改界面的设计流程与增加界面基本相同, 这里不再赘述.

1.5 删除界面设计流程

在浏览界面的中选中要删除的记录, 然后点击“删除”按钮可弹出删除提示框, 点击“确认”按钮即可完成删除. 根据所选记录的个数, 可完成批量删除和单条记录删除.

2 关键技术实现

2.1 基于 Ajax 数据交互方式

Web UI 和后台数据交互主要通过 Ajax 请求, Ajax 在浏览器与 Web 服务器之间使用异步数据传输, 浏览器每次从 Web 服务器请求少量数据, 根据实际的业务需求, 更新局部页面, 而不是整个页面[9,10].

使用 JavaScript 中的 XMLHttpRequest 对象向服务器端发送一个 http 请求, 并在回调函数中处理返回的结果. 例如, 使用 ExtJs 框架中的 Ext.Ajax.request 方法提交 Ajax 请求, 方法如下:

```
Ext.Ajax.request({
   method: 'post',
   url: ‘Corrosion.aspx’,
   params: { Method: “Select”, StrSql: “select * from water” },
   callback: function (options, success, response) {  },
   success: function (response, options) {  },
   failure: function (response, options) {  }
});
```

在 Ext.Ajax.request 方法中, method 发送 Ajax 请求使用 get 或 post 方式; url 指定要请求的服务器端 url; params 为传递的参数列表; callback 为 Ajax 请求的回调函数, 无论调用成功或失败, 该函数都会执行. 传递给回调函数的参数有三个, options 表示执行 request 方法时的参数, success 表示请求是否成功, response 表示用来执行 Ajax 请求的 XMLHttpRequest 对象. success 和 failure 为 Ajax 请求执行成功后或出现错误执行的回调函数, 传递给回调函数两个参数于 callback 相同.

Json 采用独立于语言的文本格式, 易于机器解析和生成, 可用于异步应用程序中服务器和客户机之间的数据传输. 本文 Web 服务器将数据库数据传递给客户端浏览器是通过 Json 字符串. 例如, 数据浏览时, 业务逻辑处理单元向数据访问层发送 Ajax 请求, 数据访问层通过通用数据处理单元将数据库中取出的数据放在数据集的临时表 DataTable 中, 将其转化为 Json 字符串, 转换步骤如下:

Step1. 获取 DataTable 的行数 RowCount 和列数 ColumnCount;

Step2. JsonStr="{total:" + RowCount + ",rows:[", i=0, j=0;

Step3. If(i<=RowCount);

Step4. JsonStr += "{";

Step5. If(j<=ColumnCount);

Step6. 获取 DataTable 字段列名 ColumnName，JsonStr +=ColumnName+ ":"+DataTable[i][j]; j++;转向 Step5;
Step7. JsonStr += "}"; i++;转向 Step3;
Step8. JsonStr +="]}".

2.2 大批量数据分批显示

在通过 Ajax 请求大批量数据时，为了缩短响应速度，提高数据显示效率，采用分批方式显示数据．每次在请求数据时，需要给服务器传入参数 start 和 limit，start 为加载的第几批数据，limit 为每批次显示的记录数；在后台通过 Request 的 QueryString 方法获取 start 和 limit 这两个参数查询数据，分批显示的 Sql 语句如下:

```
select top limit ID,需要显示的字段 from TableName
where (ID not in (select top limit * (start - 1) ID from TableName order by ID Desc ))
order by ID Desc;
```

其中, ID 是表 TableName 的主键，为自增序列.

3 应用实例

3.1 ExtJs 简介

本文介绍的基于轻量级数据操作架构的应用系统，在实现时可与现在主流的 Web UI 框架相结合，如 ExtJs, JQuery, Flex, EasyUI 等．本节使用 ExtJs4.0 框架介绍石化腐蚀数据操作界面的设计与实现．ExtJs 是用 Javascript 编写的库类，主要用于创建 RIA 富客户端界面，是一个与后台技术无关的前端 Ajax 框架.

3.2 腐蚀数据浏览界面

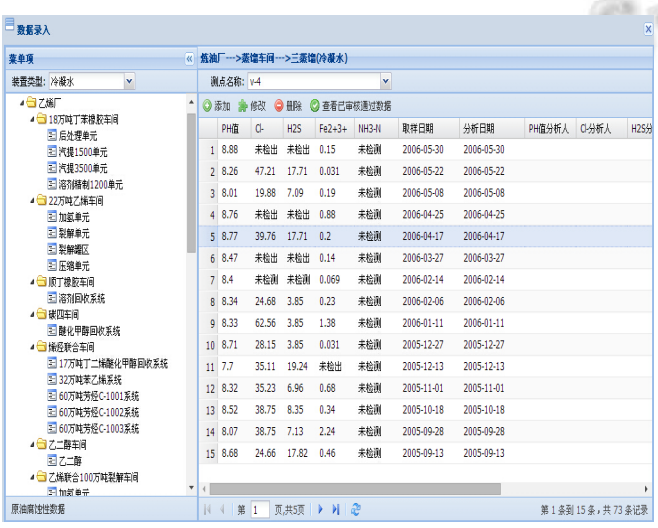

图 5　石化腐蚀数据冷凝水浏览界面

下面介绍石化腐蚀评价系统中冷凝水数据的浏览、增加、修改和删除界面的设计实现步骤．数据操作界面采用分层方式，首先呈现给用户的是冷凝水数据浏览界面，通过工具栏可完成数据增加、修改和删除．图 5 为石化腐蚀数据冷凝水浏览界面．数据浏览界面用到主要组件有 TreePanel、GridPanel，TreePanel 组件用于呈现某炼油厂工厂车间装置层级结构．GridPanel 组件用于显示测点监测到的腐蚀数据[11]，其实现步骤如下:

(1) 创建数据模型 ModelCorrosion

```
Ext.define('ModelCorrosion', {
        extend: 'Ext.data.Model', //必须继承类
        fields: [
        { name: 'ID' },        //主键，自增序号
        { name: 'PH ' },
        { name: 'CL' },
         ……//省略其他字段
]
});
```

(2) 定义数据源 StoreGrid

```
var StoreGrid = Ext.create('Ext.data.Store',
{ model: 'ModelCorrosion', //定义 store 使用的数据模型
  pageSize: 20,          //设置每页显示的记录数
  proxy: { type: 'ajax', //定义 store 的数据提交方式
  url: 'Corrosion.aspx?Method=Select',
  //请求的服务器端的地址及参数
  reader:{ type: 'Json',
  totalProperty: 'total',
  root: 'rows'} //定义数据读取方式
  }
});
```

Ext.data.Store 是 Ext 中用来进行数据交互的中间件，GridPanel 可通过它实现数据读取操作．StoreGrid 采用 Ajax 提交至后台 Corrosion.aspx 请求数据，所传参数 Method 为‘Select’，后台通过该参数可知前台需要返回浏览数据．StoreGrid 数据读取方式为 Json，其中 totalProperty 为总记录数, root 为数据读取的根节点.

(3) 定义数据显示控件 Grid

```
var Grid = Ext.create('Ext.grid.Panel',
{ width: document.body.clientWidth*0.75,
```

```
height: document.body.clientHeight,
tbar: TBar,        //定义操作工具栏
store: StoreGrid,
columns: [
{ xtype: 'rownumberer', sortable: true },
{ id: 'PH ', header: "PH 值",dataIndex: 'PH'},
{ id: 'CL', header: "CL-", dataIndex: 'CL' },
……//省略其他显示的列
]
});
```

GridPanel 组件是 ExtJs 用于显示数据最常用组件之一. 配置项 width 和 height 为 GridPanel 宽和高, 为了使 GridPanel 能适应不同分辨率的显示器, 该值可取文档体在客户端的宽度和高度, 在冷凝水浏览界面, GridPanel 宽占整个窗口的 75%, TreePanel 占 25%; 配置项 tbar 为工具栏, 定义“增加”、“修改”、“删除”等操作按钮; 配置项 store 为数据源 StoreGrid; 配置项 columns 定义 GridPanel 中显示的列, dataIndex 为列要显示的数据, 与数据模型 fields 配置项定义的字段相对应.

3.3 腐蚀数据增加界面

在图 5 所示的浏览界面上点击“增加”按钮, 弹出增加窗体, 通过弹出窗口录入腐蚀数据, 点击“保存”按钮先验证录入信息的合法性, 如果合法直接保存, 否则将提示非法信息. 图 6 为石化腐蚀数据冷凝水录入界面. 用到的主要组件有 Form 和 Window 组件, 定义弹出窗体如下:

图 6 石化腐蚀数据冷凝水增加界面

```
var Window = new Ext.window.Window({
title: ‘冷凝水数据添加’,
layout: 'fit',
width: 600,
height: 400,
modal: true,          //设置遮罩参数
items: Form,          //录入数据表单
buttons: [
{ text: "保存", handler: function () { Save(); } },
{ text: " 返 回 ", handler: function () { Window.close(); }}
]
});
```

为了保证数据操作的正确性, 弹出窗体遮罩原有浏览界面,可设置配置项 modal 为 true. 在录入表单中录入腐蚀数据, 用到的组件有下拉菜单、日期组件和输入框组件, 这些组件均定义在 Form 组件中. Form 组件的 Items 配置项可为 TextField、ComboBox、Checkbox 和 Radio 等组件. 根据录入项不同可选择合适的组件. 在石化腐蚀数据冷凝水录入界面中, 腐蚀数据 PH、CL 等录入项都用到 ComboBox 组件, 将其 editable 属性设置为 true, 这样常用值为下拉菜单值, 其他值可手工输入.

修改界面与增加界面实现过程基本相同, 这里不再赘述. 删除操作只需在删除按钮的单击事件里调用 Ajax 请求删除所选中的记录, 并用 MessageBox 组件显示提示信息.

4 结语

Web2.0 时代的到来, 使得前台用户体验变得尤为重要. 任何一个动态 Web 应用程序都包括数据操作. 数据操作界面的交互性、友好性、响应速度直接影响到软件的质量和用户的使用效果. 本文给出的解决方案, 与主流的 Web UI 框架 ExtJs, JQuery, Flex, EasyUI 等相结合, 可设计实现具有表现力、交互性强的 Web 应用程序, 在工程开发过程中具有一定的技术参考价值.

参考文献

1 李林辉,刘东,杨冬亮,等.基于 Web 计算框架的调度日报系统.电力系统自动化,2011,35(15):98−101.

2 郑迪文,沈立炜,彭鑫,等.基于 AJAX 的 Web 应用构件组装技术及工具.计算机科学,2014,41(11):152−156,191.
3 张建军,刘虎.基于 ExtJS 的 J2EE 轻量级框架的研究与应用.计算机应用与软件,2014,31(4):73−76.
4 黎吾鑫,王新.基于 Extjs+Spring MVC 的 Web 系统框架及应用研究.云南大学学报(自然科学版),2013,35(S2):110−115.
5 来天平,欧阳荣彬,王素美,等.一种轻量级企业应用 Web2.0 开发框架—Beehive+ExtJs+Json.实验技术与管理,2011,28(4):296−298,310.
6 李皎.一种 MIS 通用数据操作模板的设计与实现.中国科技信息,2010,(18):93−94.
7 卫红春.信息系统分析与设计(第二版).西安:西安电子科技大学出版社,2010.
8 刘天时.软件案例分析.北京:清华大学出版社,2008.
9 孙光明,王硕.基于 JSON 的 Ajax 数据通信快速算法.计算机应用与软件,2015,32(1):263−266.
10 Adam D, Joanna D, Sebastian C. JavaScript frameworks and Ajax applications. Communications in Computer and Information Science, 2014, 431: 57−68.
11 邓伟成,范轶翔,夏翔,等.ExtJs 框架下 Grid 组件的扩展及应用.计算机应用,2012,32(S1):80−82.