

海量数据录入的几种技术方案的比较分析

罗旭

摘要: 基于B/S结构的OA系统,受浏览器的限制,在进行数据录入和数据校验时,如何保证用户使用的方便性、实时性以及安全性是企业Web OA系统开发实践中的难点问题。对此提出了5种技术解决方案,每个方案都按照方案概述、体系流程描述、难点问题解决方法描述以及方案优缺点描述的顺序进行说明,并对5种方案从用户需求的角做了比较。

关键词: 数据录入 办公自动化 Internet

Comparison of Several Data Collection Tech for Web OA Systems

Luo Xu

(College of Traffic and Transport, Northern Jiao Tong University, Beijing 100044)

Abstract: When Designing and Developing OA (Office Automation) systems based B/S infrastructure, the data collecting and the data-verifying module are the necessary elements of the OA system. It a great difficulty to implement these two modules for the end-user with the convenience, real-time and the application security. In this paper, five practical solution and given, compared and analyzed.

KeyWords: Data Collecting, Office Automation, Internet

1 引言

随着Internet和www相关技术的发展,企业的OA系统已发生了革命性的变革。许多企业的OA系统由传统的C/S架构转到了B/S架构上。B/S结构的OA系统虽然有许多优点,也有一些固有的缺陷。浏览器端海量数据录入(以下简称海量录入问题)问题就是其中之一。

基于Internet/Intranet的Web OA系统中,大量的原始数据都要通过相关人员手工录入。也就是说,录入模块是系统中使用最为频繁,用户最为关心的模块之一。那么如何完满地解决海量录入问题就成为整个系统的核心问题,也是难点问题之一。本文结合铁道部物资总公司集采专供系统的需求和开发经验,对几种数据录入技术方案从体系流程、难点问题解决方法以及方案优缺点等几个方面进行了分析和比较,并给出了比较结果。

2 集采专供系统数据录入需求分析

2.1 数据录入的用户和主要业务

在铁路物资总公司集采专供业务活动中,需要系统用户进行数据信息录入的用户和有关业务:

(1) 铁路用户:物资需求计划、供应告急、销售合同、产品质量反馈意见和信息查询等;

(2) 总公司用户:物资需求计划、资源计划、要货计划、配置计划、质量问题处理、信息发布和信息查询等;

(3) 办事处用户:合同签订信息、运输计划和发料单等;

(4) 资源厂用户:资源信息、信息查询等。

2.2 数据录入的特点

集采专供业务是铁路物资总公司的传统业务,业务流程比较确定,对所有录入的计划报表单据都有严格的规范。系统对数据信息录入的共同特点:

(1) 数据信息量大。在集采专供系统需要进行信息数据录入的业务中,各种计划和合同管理方面数据信息录入业务及其内容较多,用户一次输入的数据信息量大。因此,要求录入数据信息的安全、可靠地传输。

(2) 录入数据信息的匹配校验。集采专供业务作为总公司的传统业务,各个铁路用户、办事处、资源厂与所在地铁路车站之间都有比较确定的对应关系。因此,要求系统具有对录入的数据进行查询匹配或者校验功能。

3 集采专供系统数据录入的难点和要求

3.1 海量录入的难点分析

主要难点在于:在海量录入时,经常要对录入的数

据进行查询匹配或者校验,如拼音码查询匹配等。其中查询匹配和校验的规则是存储在数据库中的。也就是说,整个海量录入模块至少包含数据库访问、业务逻辑处理以及 UI 逻辑处理等三个子模块。由于浏览器端的编程能力非常有限,使得浏览器端海量录入问题的解决难度加大。

3.2 海量录入模块的用户要求

综合考察集采专供的业务,现总结归纳海量录入的用户要求如下:

3.2.1 安全性

由于集采专供系统运行在 Internet 以及 Intranet 上,应用的安全以及业务的安全问题比较突出,如 Applet 小程序的安全性等。所以安全性是系统的重要需求之一。

3.2.2 便捷性

用户在使用系统的过程中,经常需要手工录入大量的业务数据。如何为用户提供一种方便、快捷的输入方式成为系统能否成功实施的核心问题之一。

3.2.3 数据校验功能

指输入有效性检验以及业务逻辑约束校验等。该功能是保证业务系统完整性、正确性的必要条件。

3.2.4 网络流量尽量小

在 Internet 范围内运行系统,由于国内带宽条件的限制,系统的通信流量要尽量小。否则系统反应过慢,导致用户无法使用。

3.2.5 通用性

指系统的跨平台能力,这里主要指浏览器端应用对各种浏览器的适应能力。集采专供系统的用户所使用的平台和浏览器有可能多种多样。为此,需要考虑系统的通用性问题。

3.2.6 向上兼容性

一般来说,许多生产企业都有比较成熟的基于 C/S 架构的办公软件,这里向上兼容性指基于 Web 的 OA 系统与原有 C/S 架构系统的操作习惯相兼容。

4 海量录入技术方案的对比分析

4.1 方案一

本方案是最普通和常见的数据录入方案。在海量录入过程中,录入项有多个,数据校验也要有多次,这意味着用户要忍受蜗牛般的录入过程。显然这种方案仅仅是一种技术上可实现的方案,但对于海量录入而言,实际价值很小。

表 1 方案简述

方案编号	浏览器端	Web 服务器端
一	HTML 页面	C、Perl 等语言写的 CGI
二	包含有 Java Applet 对象的 HTML 页面	无
三	包含有 Java Applet 对象的 HTML 页面	RMI 远程调用对象(可以在任意一台服务器上)
四	含有 VBScript 的 HTML 页面,在本地系统目录下有注册过的 COM 组件,它负责与数据库的交互	无
五	含有 JavaScript 或 VBScript 的 HTML 页面	ASP 页面,此处 ASP 泛指动态页面脚本语言,如 Microsoft 的 ASP, SUN 的 JSP 以及 PHP 等

4.2 方案二

Applet 是一种可以嵌入在 HTML 页面中的对象,由支持 Java 的浏览器解释执行,^[1]它极大地丰富了浏览器端的应用。在本方案中,Applet 负责实现 UI 逻辑、业务逻辑处理以及数据库访问。

过程 1、2: 浏览器从 Web 服务器上下载 HTML 页面以及 Applet, 并进行安全校验;

过程 3、4: Applet 初始化,与数据库连接并读取约束规则。为了减少浏览器端与数据库交互的次数,降低网络流量,在 Applet 初始化时,一些使用频率很高的规则一次性地从数据库读取到浏览器端。

过程 5、6: 用户通过浏览器中的 Applet 提供的界面录入数据;由 Applet 做 UI 逻辑处理或者业务逻辑处理,并实时提供用户反馈信息。

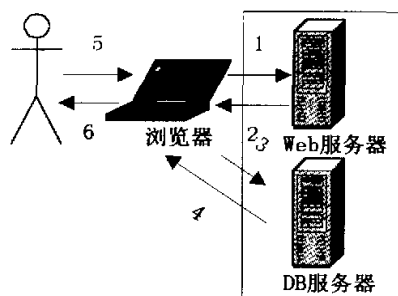


图 1 方案二和方案四的流程图

方案的优点如下: 不论是界面友好性、实时性以及可实现的功能都可与 C/S 架构的应用相媲美;借助 Visual J++ 等 IDE, 可以实现较高的开发效率、较短的开发周期。

方案的缺点如下: Applet 的安全问题比较严重。一

方面,要 Applet 具有访问数据库的能力,就必须修改浏览器端的安全策略,如配置不当,可能会受到恶意 Applet 的攻击;另一方面,Applet 是以 Java class 文件形式下载到浏览器端的,而 class 文件是可以反编译的,由此带来了较大的安全隐患;功能稍微复杂的 Applet 大小在 50K 左右,考虑到国内的带宽条件,用户在下载时,很可能会因为慢而无法忍受。

4.3 方案三

该方案与方案二有类似的地方,所不同的是引入了 RMI 技术。RMI(Remote Method Invocation)即远程方法调用,其实质是一个使用非本地对象的技术,这种技术在本方案中的作用就是把方案二中的逻辑处理模块和数据库访问模块封装到 RMI 对象中,使得 Applet 仅用来实现 UI 逻辑处理模块。方案流程图见图 2。

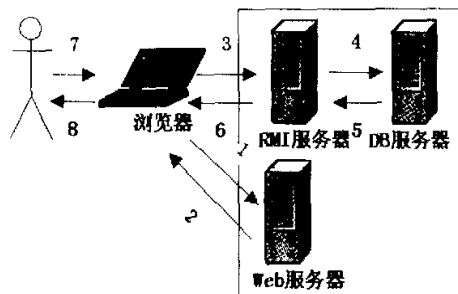


图2 方案三的流程圖

过程1、2: 浏览器从 Web 服务器上下载 HTML 页面以及 Applet, 并进行安全校验;

过程3、4、5、6: Applet 初始化, 与 RMI 连接, 由 RMI 服务从数据库中读取约束规则。同样, 为了减少 Applet 与 RMI, 以及 RMI 与数据库之间的交互次数, 降低网络流量, 提高速度, 在 Applet 初始化时, 一些使用频率很高的规则一次性地读取到浏览器端。

过程7、8: 用户通过浏览器中的 Applet 提供的界面录入数据; 由 Applet 做 UI 逻辑处理或业务逻辑处理, 并实时提供用户反馈信息。如需要获得额外的规则信息, 参见过程3、4、5、6。

方案的优点如下: 系统采用多层架构, 均衡负载; 不论是界面友好性、实时性以及可实现的功能都可与 C/S 架构的应用相媲美; 较方案二而言, Applet 变小, 页面的下载速度加快, 借助 Visual J++ 等 IDE, 可以实现较高

的开发效率、较短的开发周期。

方案的缺点如下: 受 Java 的安全机制影响, Applet 客户端与 RMI 远程对象建立连接时, 尤其是通过 Internet 建立连接时, 整体性能并没有得到提高; Applet 的相关安全隐患问题仍然存在; 兼容性差。到目前为止, 两大主流的浏览器 Internet Explorer 和 NetScape Navigator 均不支持 RMI 技术, 需要额外安装 Sun 公司提供的浏览器端 Java 虚拟机插件。

4.4 方案四

COM, 即组件对象模型^[4, 5]是以组件为发布单元的对象模型, 这种模型使各软件组件可以用一种统一的方式进行二进制级别的交互。在本方案中, 实际应用的是以 COM 为基础的技术 ActiveX。VBScript 脚本和 ActiveX 组件都属于 Microsoft 的技术, 很容易进行交互, 使增强浏览器端的功能成为可能。在流程和架构上, 本方案与方案二比较类似。区别是: 在方案二中, UI 逻辑处理和业务逻辑处理均利用 Applet 实现, 而在本方案中则利用 HTML 和 VBScript 共同实现 UI 逻辑处理, 利用 ActiveX 实现业务逻辑出来以及数据库访问模块。

过程1、2: 浏览器从 Web 服务器上下载 HTML 页面, 并下载、安装 ActiveX 组件。(注: ActiveX 在第一次使用需要下载, 以后则不再需要);

过程3、4: ActiveX 组件初始化, 与数据库连接并读取约束规则。为了减少浏览器端与数据库交互的次数, 降低网络流量, 在 ActiveX 组件初始化时, 一些使用频率很高的规则一次性地从数据库读取到浏览器端。

过程5、6: 用户通过浏览器中的 Form 表单提供的界面录入数据; 由 VBScript 调用 ActiveX 组件提供的接口做 UI 逻辑处理或业务逻辑处理, 并实时提供用户反馈信息。

该方案的优点如下:

借助 InterDev、Visual Basic 等 IDE, 可以实现较高的开发效率、较短的开发周期;

该方案的缺点如下: 通用性下降, 应用的平台仅限于 Microsoft 的 Windows 操作系统和 Internet Explorer

4.5 方案五

VBScript 和 JavaScript 都是由浏览器解释执行的脚本语言, 不同的是 JavaScript 的通用性更好。许多主流浏览器都支持 JavaScript 技术, 如 Internet Explorer、Netscape Navigator 等, 而仅有 Internet Explorer 支持 VBScript。所以在此仅以 JavaScript 为例进行讨论。本

方案中由HTML、JavaScript共同实现UI逻辑处理,由JavaScript实现业务逻辑处理,由动态脚本技术实现数据库访问模块。

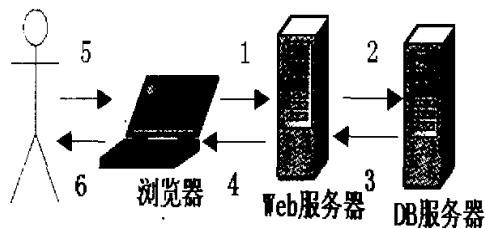


图3 方案五的流程

过程1:浏览器向Web服务器请求动态脚本生成的HTML页面;

过程2、3、4:在Web服务器端,利用动态脚本技术访问数据库,读取规则数据,并动态生成包含规则数据的HTML页面,最后由Web服务器把HTML页面发送到浏览器;

过程5、6:用户通过浏览器中的Form表单录入数据,并利用JavaScript脚本实现UI逻辑处理和业务逻辑处理,并用用户以信息反馈。

在此方案中,难点问题的解决过程是,首先规则数据由ASP脚本从数据库中获取,在生成HTML页面的同时把规则数据包含在HTML页面中,也即保存在浏览器端,利用JavaScript的强大功能实现拼音码的匹配以及数据校验。方案的优点:平台通用性强,由于在浏览器端仅需支持HTML页面和JavaScript脚本的解释,所以包括Internet Explorer和NetScape Navigator在内的很多浏览器都可以做为运行时平台;不论是界面友好性、实时性以及可实现的功能都接近C/S架构的应用;速度快。由于规则数据一般不超过30k,所以页面下载速度比较快;UI逻辑处理和数据校验都在浏览器端完成,系统响应迅速;兼容性较好,系统兼容多种浏览器以及Web服务器。

方案的缺点:到目前为止,还没有一款象InterDev一样强大易用的IDE环境支持JavaScript的调试开发,所以导致开发效率降低。

到此5种解决方案都已经介绍完毕。下面结合集采专供系统中海量录入的需求对5种方案做一比较。

表2 各个方案满足需求情况对比表

需求 \ 方案	方案一	方案二	方案三	方案四	方案五
安全性	是	否	否	是	是
便捷性	否	是	是	是	是
数据校验功能	是	是	是	是	是
网络流量尽量小	否	否	否	是	是
平台通用性	是	否	否	否	是
向上兼容性	否	是	是	是	是

5 结束语

通过比较可知方案五是最优方案。它不但完全满足了各种需求,而且采用了目前Web开发中最为流行的动态脚本生成技术,具有较高的实用价值。虽然本文给出的技术方案主要是结合铁道部物资总公司集采专供系统的需求,但是它对其他Web OA系统海量录入问题的解决同样具有很大的实用价值。在实际开发过程中,通过比较分析,第五个技术方案被选中用于实现Web方式下的海量录入模块。目前,集采专供系统中的油品子系统已经投入使用。由于数据录入模块确实为用户提供了安全、简便的录入方式,因而获得了用户的一致好评。

6 参考文献

- 1 Ayers, Dz Java服务器高级编程.曾国平.北京:机械工业出版社,2000
- 2 David Flanagan. JavaScript权威指南.张铭泽.北京:中国电力出版社,2001
- 3 刘中宏.蔡锋.精通VBScript.北京:清华大学出版社,2001
- 4 Box, D. COM本质论.潘爱民.北京:中国电力出版社,2001
- 5 Matt J. Crouch.怎样用ASP和COM进行Web编程.康博创作室.北京:人民邮电出版社,2001
- 6 廖彬山,高峰霞,徐颖. Web数据库开发技术.北京:电子工业出版社,2001
- 7 Ueli, Wahl等. WebSphere环境下Servlet与JSP高级编程.翟裕忠,张剑峰,王丛刚等.北京:机械工业出版社,2001
- 8 Garfinkel, S. Web安全与电子商务.何建辉等.北京:中国电力出版社,2001
- 9 李明武,郭胜,夏帮贵,吉雷. JavaScript&HTML精通网页编程.北京:北京希望电子出版社,2001

(收稿日期2001-09-11)