Struts国际化实现小语种考试系统

党小超,马 威
(西北师范大学 网络教育学院,甘肃 兰州 730070)

摘 要:本文在研究程序国际化与 Struts 框架结构的基础上,给出了 Struts 架构程序国际化实现模型,通过分析该模型在解决国际化过程中产生乱码的原因,给出了选择合适的字符集、消除文件中的硬编码等方法,并详细阐述了 Struts 架构程序国际化的具体实现方法。

关键词:国际化 Struts 框架 Java 小语种考试
中图分类号:TP311 文献标识码:B 文章编号:1673-8454(2008)09-0062-03

目前,小语种考试的字符编码问题阻碍了网络考试系统在外语考试中的应用,Struts 国际化编程方法能很好地解决这个问题。程序国际化是指把计算机系统或应用软件改写为同时支持多种语言和文化习俗的过程。为了快速开发完成多语言系统就要解决如何把与编码有关的元素、标题、图片链接、提示等在程序之外存储;如何依据不同的用户,选择加载不同的语言种类以及如何共享多种语言的数据等问题。

小语种网络考试系统在考试系统设计开发的过程中面临三个问题:(1)小语种语言在浏览器中不能正确显示常出现乱码;(2)语言不同使用字符集也不同,不同的小语种试题能否一起存储;(3)系统使用统一的中文提示还是不同语种使用不同语种的提示。

为了解决上面的三个问题,我们首先就要考虑是建立一套统一的系统还是建立多套不同语种的系统,下面来看看 Struts 是如何具体解决这些问题的。

一、Struts 框架结构

Struts 是一个 MVC 框架,像 Java 和其他 Java 框架一样,Struts 能够国际化。Struts 模型层是由 JavaBean 和 EJB 构成,控制器是通过 Action 和 ActionServlet 来实现,视图层由 JSP 文件构成。

视图层中通过 Struts 提供了 Java 类 org.apache.Struts.action.ActionForm,Java 开发者将该类细分来创建表单 bean。在运行时,当 JSP 对相关的 HTML 表单进行显示时,JSP 将访问该 bean(保存要放入表单中的值),这些值是由业务逻辑或者是通过先前的用户输入来提供的。当从 Web 浏览器中返回用户输入时,该 bean 将验证并保存该输入以供业务逻辑或后续重新显示(如果验证失败的话)使用。通过 bean 标签可以直接读取 Application.properties 中的资源。

模型层中 Struts 不直接有助于模型开发,系统模型的状态主要由 ActiomForm Bean 和值对象体现。[1]

控制层中主要是 ActionServlet,但是对于业务逻辑的操作则主要由 Action、ActionMapping、ActionForward 这几个组件协调完成(也许这几个组件应该划分到模型中的业务逻辑一块)。其中,Action 扮演了真正的控制逻辑的实现者,而 ActionMapping 和 ActionForward 则指定了不同业务逻辑或流程的运行方向。[2]

此外我们还引入了 Hibernate 来管理数据库,通过 Hibernate 把数据库封装成对象,在 Struts 中操作数据库就可以像操作 Java 对象一样来完成,由于数据库采用了 Hibernate,所以用户在操作数据库时不必考虑是什么数据库,只需在 Hibernate.xml 中配置数据库 jdbc 驱动或者数据库连接池以及 sql 语句翻译类,而在数据库移植的过程中也只需要修改 Hibernate.xml。图 1 是 Struts+Hibernate 框架的组件结构图。

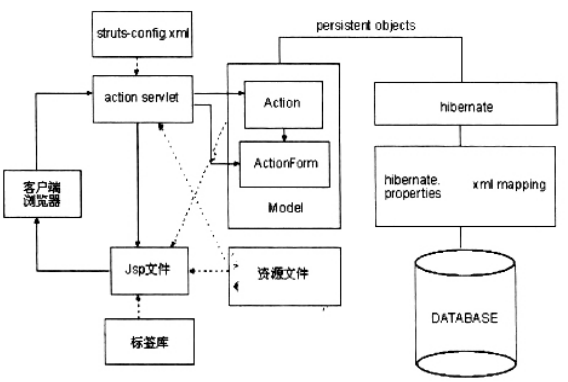

图 1 Struts+Hibernate 框架的组件结构图

如图 1 所示我们把整个 Struts 结构分为三大部分,第一部分由浏览器、JSP 文件和 Action Servlet 组成。这部分主要是用户界面,负责数据显示以及数据提交,在这里没有任何的逻辑操作,用户可以看到的就是这一层,所以国际化在一定程度上就是在这一层输出符合用户习惯的语言。第二部分包含 Action 和 Action Form 两部分,它们共同来完成整个系统的业务逻辑。第三部分就是 Hibernate 生成的数据库对象。

为了实现国际化，可以根据上面的原则把图 1 演化为图 2。

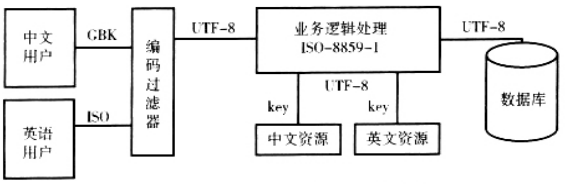

图 2 Struts 国际化实现模型

了解了 Struts 结构，以及 Struts 国际化实现模型后，就要决定是建立一套同时支持多语种的系统，还是建立多套不同语种的系统，也就是使用 i18n(Internationalization 国际化)，还是 i10n(localization 本地化)的问题，国际化程序能够支持不同的语言以及不同格式的日期、时间、货币和其它值，而无需软件修改。这通常涉及‘软编码’或将文本组件同程序代码分离并且可能涉及可插入代码模块；本地化是设计和编写能够处理特定区域、国家或地区、语言、文化、企业或政治环境的应用程序的过程。从某种意义上说，为特定地区编写的所有应用程序都本地化了，虽然这些应用程序大多数只支持一种语言环境。然而，真正的本地化通常是由访问语言环境、位置、政治或其它特定组件和模块的核心代码，以及将文本翻译成适合于用户的版本来实现的，也就是采用本地化的程序编写方法需要对每个语种的平台核心代码中的文本内容都进行翻译，新加一个语种就进行一次程序翻译，而国际化不需要进行程序翻译，只需要翻译资源文件就可以了，那么对于编写多语种的考试系统采用国际化会容易很多，也就是说我们需要建立一套支持国际化的程序。下面来看看如何具体解决前面提出的三个问题。

二、Struts 国际化实现

1.乱码问题

小语种语言在浏览器中不能正确显示常出现乱码的原因是计算机不能正确选择不同语种的编码和解码方法而造成的。字符是计算机表达信息的主要方式，字符的主体部分是 ASCII，ASCII 是一个七位的编码标准，包括可打印符号、控制符号等。由于计算机通常用‘字节(byte)’这个八位的存储单位来进行信息交换，因此不同的计算机厂家对 ASCII 进行了扩充，对值大于 127 的 128 个符号予以定义，并赋予符号的形状，这些‘扩展的 ASCII’字符只有在特定的环境下才具有‘交换’的意义。[3] 计算机以及很多计算机网络协议的制定都是建立在 ASCII 码的基础上，但是 ASCII 码用于计算机信息的表示有很大的不足，主要表现在多国文字、图形、声音等二进制文件、信息压缩、信息保密等许多方面。因此，在 ASCII 和扩展 ASCII 码的基础上，用一定的规则定义一些新的信息表达形式，就形成了信息传输和处理中的一大类概念和事物，即编码和解码。当信息编码和解码能够统一的时候，信息是可以被交换和理解的；相反，当信息编码和解码不能够统一的时候，信息就不能被交换和理解，这就是乱码。乱码的产生既然是信息编码和解码不能够统一的结果，那么，解决乱码的过程就是找到和编码相统一的解码方法，并对计算机软件中不能全自动进行适当解码的信息进行重新的处理和解码，使得恢复信息可以被理解和交换。

实现程序的国际化就是告诉计算机，针对不同的用户应该使用哪种编码与解码方式。根据数据的操作流程，在三个不同阶段的操作可能会出现乱码问题：(1)数据库中的数据如何正确编码解码；(2)Java 程序中出现的数据如何正确处理；(3) 客户端输入输出的信息如何正确处理。可以对这三个阶段划分层次，如图 3 所示。

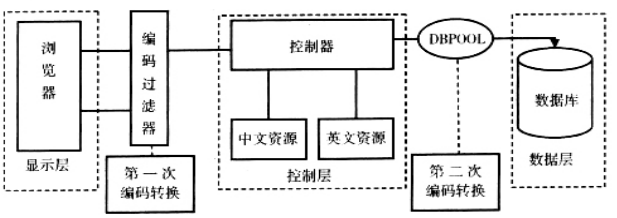

图 3 编码转换图

因为 Java 中采用的编码格式是 ISO8859-1，所以可以把数据库的存储格式选择为 ISO8859-1，也可以是 UTF-8 格式，这样在数据库到操作层转换的过程，就可以在 Hibernate.xml 中进行配置转换，只要能保证编码、解码的一致性，就可以确保在数据操作层和数据存储层之间不出现乱码。我们有两种实现方法：

(1)对每条输出的数据都进行编码和解码

为保证向客户端输出时采用特定的编码方式，首先在 JSP 源代码中加入下面一行：

<%@page contentType="text/html; charset=特定编码"%>

特定编码可以是 GBK、BIG、GB2312 等。

为了能正确获得传入的参数，在 JSP 源文件头加入下面一句：

<%request.setCharacterEncoding("特定编码");%>

为了让 JSP 编译器能正确地解码含有特定编码字符的 JSP 文件，需要在 JSP 源文件中指定我们的 JSP 源文件的编码格式，具体来说，在 JSP 源文件头上加入下面的一句即可：

<%@page pageEncoding="特定编码"%>

(2)在输入、输出的过程中采用过滤器

使用这种方法只需要对编码编写 Filter 文件，并且在 WEB.XML 中进行配置就可实现。

Servlets Filter是Servlet 2.3规范中新增加的，它截取用户从客户端提交的请求，在还没有到达需要访问的资源时运行的一个类。它操纵来自客户端的请求，在资源还没有发送到客户端前截取响应，并处理这些还没有发送到客户端的响应，Filter的特点刚好满足数据显示层和数据操作层之间对字符编码进行转换的要求。

通过Filter就可以确保数据从客户端向数据操作层开始输入数据，整个流程都采用同一种编码，直到输出都不必考虑编码格式，只需要配置过滤器就可以完成，而不需要对每页都进行字符转换。

对于不同的语言环境，用户可以选择本地化的编码程序进行浏览，增加一种新的语言，只需做两件事，翻译程序中出现的硬编码和配置适合该编码的编码过滤器。以上手段就可以解决如何正确编码和正确解码，使程序不出现乱码，从用户使用的角度来说就实现了程序国际化。

Struts编写的国际化程序具有良好的移植性，在不修改任何代码的情况下通过增加资源文件增加新的语种。

2.选择合适的字符集

语言不同使用字符集也不同，不同的小语种试题要存储在一起就要使用能兼容各语种的字符集。相同的字符，编码也相同，这样在检索的过程中就不会出现相同的字符有不同编码显示的问题了[1]。可以选择一种统一的编码，可供选择的大字符集编码有：(1)UNICODE双字节字符集(ISO8859-1)；(2)UTF-8。UNICODE的所有字符（包括英文）都用2个字节表示，ISO8859-1英文都用2个字节存取，空间浪费太多，传输量增大。而UTF-8，UNICODE的2字节字符用可变长类型(1-3个字节)表示，对英文，仍然和ASCII一样用1个字节表示，这个字节的值小于128，对其他语言用一个值位于128-256之间的字节开始，再加后面紧跟的2个字节表示，一个字符共3个字节，因此，在程序处理过程中所有字符都是16位(双字节)，但在存取转换成字节流时使用UTF-8格式转换，英文字符与ASCII方式存取时大小是一样的，而且UTF-8的一个特别的好处是它与ISO8859-1完全兼容，所以选择UTF-8。

3.消除文件中的硬编码

系统使用统一的中文提示还是不同语种使用不同语种的提示，这个问题实际就是消除硬编码的问题。硬编码常常出现在许多程序中。在开发程序的时候，开发人员有时候图方便，赶进度或者想不到更好的方法，只好将某些业务逻辑，图片文字资源固定死，即把某一种语言资源直接写入程序代码中。要彻底解决国际化问题就一定要消除硬编码，由用户浏览器语言环境或用户选择来加载不同的语言资源。通过key加载和Locale相符合的文本和图片，Struts框架把用户的Locale实例保存在session范围内，这样，Struts框架能自动根据这一Locale实例从Resource Bundle中选择合适的资源文件。视图层对Resource Bundle的操作可以通过<bean:message>标签简单、高效地实现。<bean:message>标签的工作原理是，根据Struts-config.xml中定义的资源文件，Application.properties，到WEB-INF/classes/resource/去找Application.properties文件，从以上配置信息的表面上看起来是这样，但通过查看Struts的源码，可以看到：Application.properties不能随便修改.properties扩展名，因为Struts并不是单纯去寻找Application.properties文件，而是首先找到Application，赋给name变量，然后加上下划线“_”，再加上localeKey(如zh，en)，再加上.properties。确定了文件名之后，Struts使用了ClassLoader类的getResourceAsStream方法得到了一个InputStream，然后Struts使用了Java.util.Properties类的load方法，将资源文件中的所有资源读出放到一个HashMap里面，Struts就可以根据key值取出不同的message给视图层。通过<bean:message>读取.properties文件就可以看到系统是把资源文件加载到HashMap中，所以系统运行速度会非常快。[5] 这样在系统第一次运行的过程中生成class文件里所有的硬编码都转换成了${key}的形式，在最后输出的时候替换成具体的文本、图片、按钮等，这样就消除了整个程序中的硬编码。所有硬编码内容都转换成了软编码存储在Application_xx.properties中，由于Java采用ISO8859-1编码，所以需要使用native2ascii把Application_xx.properties转成Java所需的ISO8859-1编码。

三、结束语

Struts国际化编程实现小语种网络考试系统，业务逻辑的实现和普通网络考试系统一致，不同之处是小语种考试系统通过统一输入输出编码，使用大字符集代替本地化字符集，消除程序中的硬编码，实现适用于小语种考试的网络考试系统，使网络考试系统的优越性在外语考试中得以充分发挥。◉

参考文献：

[1]陶克，王东，刘尔才.基于J2EE架构的设备能源管理的企业资源计划(ERP)系统研究[J].科学技术与工程，2005(11).

[2]Malcolm G. Davis. Struts, an open-source MVC implementation.http://www-128.ibm.com/developerworks/library/j-Struts/index.html,2001,2

[3]华继钊.网络中不规范字符的处理[J].通信世界,1999(10).

[4]孟照星等.Java程序国际化字符处理研究[J].计算机应用,2003(23).

[5]Sing Li. Create internationalized JSP Applications.http://www-128.ibm.com/developerworks/java/library/j-jspapp/,2005.5.