

Struts 框架的扩展与改进

叶娜^{1,2}

(1. 西安交通大学 电信学院, 陕西 西安 710049;

2. 西安建筑科技大学 信控学院, 陕西, 西安 710055)

摘要: Struts 框架是目前流行的 Web 应用开发框架之一。为了更好地使用和开发框架, 在对 Struts 框架源码进行深入分析的基础上, 总结了 Struts 框架中的主要扩展点, 包括插件的使用以及配置类和控制器的扩展点; 结合实际使用经验, 提出通过制定基 Action 方法模板以便于开发。此外, 提出视图和模型机制的扩展和改进方案, 包括 ActionForm 子类代码自动生成、Struts 标签库的扩展以及使用业务委托和服务定位器设计模式访问模型。

关键词: Struts; 框架; 控制器

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2007)12-0016-04

Extention and Improvement of Struts Framework

YE Na^{1,2}

(1. Telecommunication School, Xi'an Jiaotong University, Xi'an 710049, China;

2. School of Information and Control, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: Struts is one of the popular frameworks for Web application development. In order to use and develop framework better, the extension points of Struts were summed up in this paper based on deep analysis on the source code of it. The main extension points included the plugin and the extension to the configuration class and the controller. According to practical experiences, the base Action methods template was proposed for easily developing. Furthermore the extension and improvement proposals of the view and model including the automatic code generation of the ActionForm subclass, the extension to Struts taglib and accessing the model component by using business delegate and service locator design patterns were raised.

Key words: Struts; framework; controller

0 引言

如今, 在进行 Web 应用开发时, 大多数系统都使用了一些当今流行的框架, Struts 就是其中之一。Struts^[1]是 Apache 软件组织负责开发的一个开源项目, 是 MVC 设计模式/Model2 架构的一个实现, 它将应用分为模型、视图和控制器三部分。Struts 提供了自己的控制器组件, 在视图组件上主要使用 JSP 技术, 也可以和其他表示层框架如 velocity、XSLT 等结合; 在模型组件上, 不但可以使用一些标准的数据访问方式, 如 JDBC 和 EJB, 也可以和一些第三方软件包如 Hibernate, iBATIS, ORB 等结合, 是开发大规模 Web 应用程序的设计利器。

文中对 Struts1.2 源码做了深入研究, 对该框架中的扩展点进行了总结, 同时提出几点改进思路。

1 框架

1.1 框架的概念

框架是面向对象系统可以获得的最大复用方式, 较大的系统可由多层彼此合作的框架组成。框架具有如下特点:

首先, 框架封装了一组类以及类之间的协助关系, 即控制流程;

其次, 框架是某类特定应用的半成品, 是一个能被许多应用程序所采用的一个程序模板, 它不仅提供了这类应用公有的部分, 而且为应用预留了可供扩展的部分, 以满足特殊需要;

最后, 框架提供了设计和代码的复用。在开发中, 一类相似的应用共享了框架的公用部分, 这相当于对代码的复用; 同时, 在开发这些应用时, 为了有效地使用和扩展框架, 使用者必须了解框架的控制结构和设计决策, 而这则是对设计和分析的复用。

1.2 使用框架的优势

框架提供了建立复杂系统的强有力的工具, 它具

收稿日期: 2007-06-04

作者简介: 叶娜(1979-), 女, 陕西西安人, 博士研究生, 助教, 研究方向为分布式数据库、数据挖掘、Web 服务。

有如下优点^[2]:

(1)可以复用框架的体系结构和代码,减少了编码、调试和测试的工作量。

(2)使用户能够专注于业务逻辑,不需要花大量时间去做大量基础开发工作。

(3)为组合软件开发奠定了基础。设计良好的框架将允许第三方软件公司提供部分或者完整组件供用户或系统集成商装配。

1.3 框架与其他概念的关系

在基于框架的软件开发中,有一些其它的技术与框架密切相关,包括架构、设计模式、类库和组件^[3,4]。从面向对象的角度来审视它们之间的关系,可以用图 1 来表示。

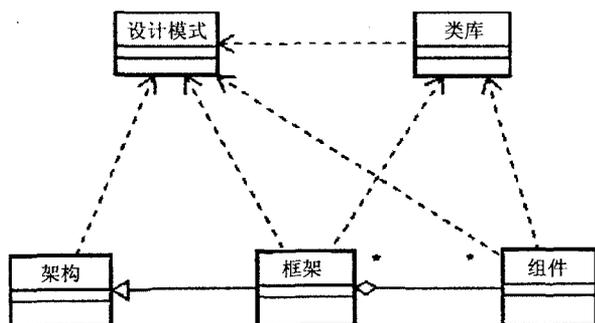


图 1 关系图

架构就如同一个抽象类,它仅对系统结构作了一个抽象的描述,而没有具体去实现。框架则是基于架构的可执行的代码,可以将它看成是对架构的具体实现。架构是框架的泛化,框架是架构的特化。框架可以通过集成若干个组件来建立,框架与组件之间是共享聚集的关系。框架和组件的实现中,都可以调用类库中的类的方法,因此,框架和组件都依赖于类库。

设计模式可应用于软件开发的各个环节,如设计系统结构、组件交互及编程实现。因此,架构、框架、组件、类库都依赖于设计模式。

2 Struts 框架的扩展点及改进

由上可见,使用框架的一个最大的好处就是能够根据具体应用的需要来扩展和定制框架。Struts 框架也不例外,它提供了几个重要的扩展点供开发者扩展。本节中总结了 Struts 的几个重要的扩展点,同时,针对 Struts 框架与业务层或持久层结合使用的问题,提出一些改进措施。

2.1 使用插件 Plug-in 机制

Struts 框架通过接口 org.apache.struts.action.PlugIn 提供了插入和自动加载组件的机制,任何 Java 类只要实现了该接口就可以作为插件使用。PlugIn 接口

包含两个方法,如下所示:

```

public interface PlugIn {
    public void init (ActionServlet servlet, ApplicationConfig config)
        throws ServletException;
    public void destroy();
}

```

2.2 Struts 框架配置类的扩展

org.apache.struts.config 包中存放了 Struts 配置文件中包含的所有配置元素所对应的类。这些配置元素大部分都有一个 className 属性,通过它可以为该配置元素提供一个 Java 类名,这使得用户可以定制配置元素或传入额外信息^[5]。通过这种机制,Struts 配置文件元素的可扩展为 Struts 框架的使用提供了极大的可扩展性及灵活性。

2.3 控制器的扩展点

2.3.1 ActionServlet 的扩展

当 Struts 应用第一次加载时,所有的初始化操作全在 ActionServlet 中进行。如果需要修改框架的初始化方式,就可以对 ActionServlet 类进行扩展,创建一个它的子类,覆盖需要修改的方法^[5]。然后在 web.xml 文件中进行指定。

2.3.2 RequestProcessor 的扩展

从 Struts1.1 开始,所有的运行期请求处理操作都移到了 RequestProcessor 类中。如果需要改变对请求的处理方式,可以扩展 RequestProcessor 类。要覆盖 RequestProcessor 的方法,首先必须能够让框架知道要使用的是用户自己指定的类,这可以在 Struts 配置文件中通过 controller 元素进行配置,如下:

```

<controller
    contentType = "text/html; charset = UTF-8"
    debug = "3"
    locale = "true"
    nocache = "true"
    processorClass = "com.oreilly.struts.framework.CustomRequestProcessor"/>

```

在 RequestProcessor 类中有一个留给用户扩展的方法 processPreprocess(),对于每个请求这个方法都会被调用,但不作任何操作,方法定义如下:

```

protected boolean processPreprocess ( HttpServletRequest request,
    HttpServletResponse response) {
    return (true);
}

```

该方法在处理 ActionForm 及执行 Action 的 execute()方法前被调用,默认返回 true,含义是告诉 RequestProcessor 继续处理该请求。用户可以使用这个方法来改变默认的请求处理行为,比如可以进行一些条

件逻辑判断, 如果不符合条件, 可以让该方法返回 false, 通知 RequestProcessor 停止处理该请求并返回到 doGet() 或 doPost() 调用, 这时用户必须负责将请求导航到某处^[5]。

2.3.3 基 Action 的扩展

(1) 基 Action 中封装公共逻辑。

在实际应用中, 经常使用一种技术, 即创建一个基动作类 BaseAction 继承于 Action, 让它作为所有其它动作类的父类。这样做的原因是在很多应用中绝大多数动作类中都包含有一些公共逻辑, 将这些公共逻辑代码移到基动作类中, 可以消除冗余^[5]。

(2) 制定基 Action 方法模板。

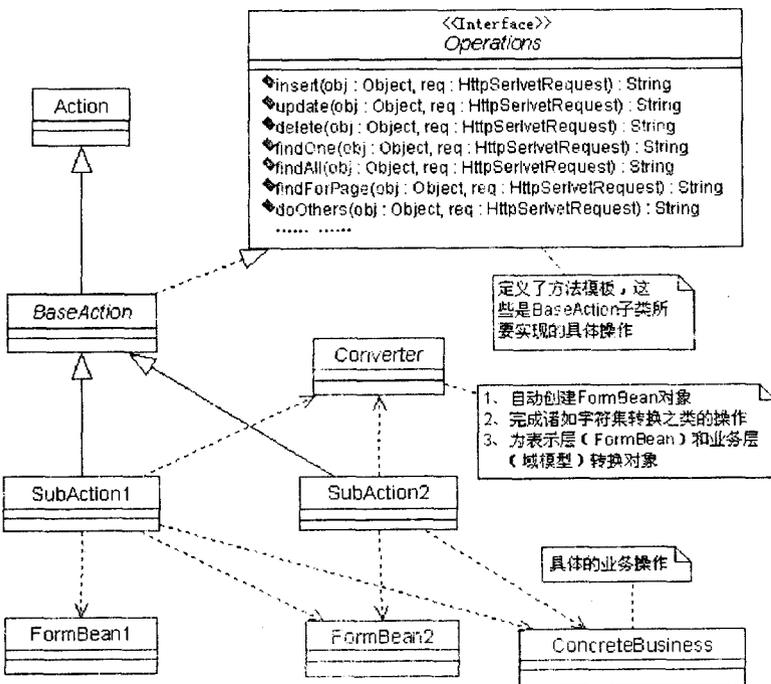


图 2 基 Action 方法模板

图 2 是 Struts 中 Action 的扩展模板。图中 Operations 接口表达的不是具体业务, 而是一些公共的操作。方法的返回值是 String, 表示通过 Action 处理后, 返回页面是什么(在 Struts 配置文件中配置)。所有方法都有一个参数 obj, 一般而言它是 FormBean 对象, 是要操作的表单数据的封装。另一个参数 req 主要负责将操作完成后所要返回的数据保存到 obj 中, 一般通过 req.setAttribute("...", Object) 方式或存放到 session 中, 另外也可以从 req 中获取其他参数值。

BaseAction 主要是把 Struts 的 Action 和这些操作 Operations 汇集到一起, 它主要实现了 Action 的 execute 方法, 它根据 request 中的条件值(flag 值, 这些条件是事先约定好的), 判断调用哪个方法(insert, update 等等, 或它们的组合)。再根据这些方法的返回字符串(下一个网页), 执行页面的跳转工作。这个类的另一

个作用是完成了一些共有操作, 如在分页处理中, 保存当前页面, 每一页的记录数, 计算下一页的页码等。

BaseAction 的代码可能会是如下所示:

```
public class BaseAction extends Action implements Operations {
    public ActionForward execute(ActionMapping actionMapping,
        ActionForm actionForm,
        HttpServletRequest servletRequest,
        HttpServletResponse servletResponse) {
        String flag = servletRequest.getParameter("flag");
        if("insert".equals(flag)) {
            ..... //若干其他操作
            return insert(obj, servletRequest);
        }
        else if("update".equals(flag)) {
            ..... //若干其他操作
            return update(obj, servletRequest);
        }
        .....
        else {
            return doOther(obj, ServletRequest);
        }
        ..... //诸如分页处理之类的公共方法
    }
}
```

BaseAction 并没有实现 Operations 里面的全部方法(如 insert, update 等), 这些方法是由 BaseAction 的子类来实现。BaseAction 的子类 SubAction 里面处理 FormBean 的数据, 通过 Converter 类进行数据转换, 调用业务类提供的方法。另外, 通过借助 Converter 类, 可以实现多个 FormBean 和一个 SubAction 相关联。

2.4 视图机制的扩展与改进

2.4.1 ActionForm 子类代码自动生成

ActionForm 子类的主要作用是接收用户输入的数据, 作为数值对象在各层直接传输, 将处理结果返回给客户端页面进行显示。ActionForm 子类中包含了各个成员属性的访问子, 即大量的 get() 和 set() 方法, 如果能够参照页面或数据库表, 自动生成 ActionForm 的子类, 将会大大提高开发人员的工作效率。

2.4.2 Struts 标签库的扩展

Struts 对页面数据的校验都是在 ActionForm 的 validate() 方法中进行(服务器端), 而不能使用 Struts 标签配合 JavaScript 在客户端校验。这样会引起如下的问题:

1) 用户每次提交的数据都传输到服务器进行验证, 之后把验证的结果返回给客户端, 将会大大增加网

络传输的成本开销。

2) 大量校验操作都由服务器完成会消耗服务器资源,现代微机(客户端)性能不断提升,部分数据校验功能在客户端完成,适当减轻服务器负担,会提升系统整体性能,这是 Rich Client 思想的一个体现。

3) Struts 标签库中对某一指定的标签属性是具有严格限制的,如 html:text 标签,如果写成 `<html:text property="productCount" title="产品数量必须为整数!" validator="isDigit"></html:text>`,在编译时就会出错,因为这个标签未提供 validator 属性。而对于 html 语言, `<input name="productCount" type="text" title="产品数量必须为整数!" validator="isDigit"/>` 却是一个正确的表示。还可以配合 JavaScript,在页面提交时扫描 validator,在客户端进行数据校验。由此可见,对 Struts 标签库进行扩展,使得能够添加用户需要的标签属性是有必要的。只有这样,才能实现上述所说的客户端页面校验。客户端数据校验对用户是可见的,用户甚至可以修改验证规则,因此它是不安全的。对于重要数据的校验,可以使用客户端及服务器的双重校验,以增强系统的安全性。

2.5 模型组件访问机制的扩展

当 Struts 框架与业务层或持久层交互时,为了保护表示层,尽量将表示层与业务层分离开,可以使用业务委托(Business Delegate)设计模式或服务定位器(Service Locator)模式访问模型。

3 对开发和选择框架时的一些个人见解

框架作为一种可复用软件,和大多数软件一样,使用者希望能应用于各种场合,希望基于框架所开发的应用程序容易维护,越“傻瓜”越好。对开发者来说,用户的需求是必须要考虑的,除此以外,框架本身的发展、进步和维护也是不容忽视的方面。由于这些原因,框架的开发变得格外复杂。为了降低框架的设计难度,应注意以下方面:

1) 框架是一种软件,框架的开发过程和软件的开发过程一样,是一个不断积累、改进和不断完善的过程。

程。建立框架时,应充分考虑其适应性和可扩展性。

2) 经常参考一些成功的框架来吸取经验,多复用一些成熟的组件,这样就会使后续的开发工作更简单。

3) 可以多参考一些设计模式。良好的设计模式可降低框架设计实现的难度,并且可更好地文档化。

4) 在框架的发展过程中,要经常使用,通过它开发一些应用程序。只有不断地使用,才能不断地发现问题,从而解决问题,使框架日趋完善。

框架的目的是被复用,以减少建立一个新系统的工作量。框架能使应用程序的开发简单,价格低廉,但只有当理解一个框架的代价少于建立一个新系统的代价时,框架的复用才成为可能。在选择框架时,应该根据项目的规模和开发团队人员的技术状况来定,尽可能地使用框架,但不能依赖框架,不能纯粹为了追求项目的技术含量,使得框架的使用成了一个美丽的谎言。

4 结束语

基于框架的软件开发,是目前很受重视的一个研究方向,也是较为普遍的软件开发方法。为了更有效地利用 Struts 框架,文中在深入剖析 Struts 源码的基础上,对该框架预留的扩展点做了总结,比如插件 Plug-in, ActionServlet 的扩展等。同时根据实际使用经验,提出了几点对 Struts 框架的改进思路,比如建立基 Action 类方法模板、使用业务委托模式和服务定位器模式访问模型等。

参考文献:

- [1] 陈 玲. 基于 Struts 框架的现代物流管理系统[J]. 中国科技论文在线, 2006(9): 1-2.
- [2] 冯玉琳, 黄 涛, 金蓓弘. 网络分布计算和软件工程[M]. 北京: 科学出版社, 2004.
- [3] 熊 江. 面向对象的框架设计[J]. 西南师范大学学报, 2003, 28(3): 391-394.
- [4] 周警伟, 罗晓沛. 实施一个面向对象框架的方法[J]. 计算机仿真, 2002, 19(3): 107-108.
- [5] Cavaness C. Jakarta Struts[M]. [s. l.]: O'Reilly and Associates Inc., 2002.

(上接第 15 页)

参考文献:

- [1] Bakke B. Simulating SCI and SCI/RT in Simula[D]. USA: Department of Information, The Institute of Electrical and Electronics Engineers, University of Oslo, 1995.
- [2] IEEE Standard For Low-Voltage Differential Signals (LVDS) For Scalable Coherent Interface[S]. IEEE std 1596. 3-1996.

- [3] IEEE Standard for Scalable Coherent Interface(SCI)[S]. IEEE Std 1596-1992.
- [4] 周 强, 罗志强. SCI 协议标准综述[J]. 航空电子技术, 2001, 32(2): 9-18.
- [5] 蔡敏芳, 陆 达. SCI 协议在高性能集群计算中的可行性分析[J]. 计算机科学与技术, 2005(8): 4-8.
- [6] IEEE Standard For Shared-Data Formats Optimized For Scalable Coherent Interface[S]. IEEE std 1596. 5-1993.