

文章编号: 1000-5013(2009)05-0526-04

引入 intelliWeb 的 iBAF 应用框架设计与实现

卢晓星, 余金山

(华侨大学 计算机科学与技术学院, 福建 泉州 362021)

摘要: 为了降低 J2EE 的开发复杂度并缩短软件开发周期, 设计并实现了一种基于 intelliWeb 的应用框架 iBAF. 该框架在 Web 层、业务逻辑层和持久层中, 分别引入 intelliWeb、业务管理公共核心设施和数据管理公共核心设施, 为二次开发提供一些稳定的基础服务和模块. 在实现原有的核心代码与第 3 方产品之间胶合层的过程中, 引入业务管理公共核心设施, 使得胶合层更加轻量、灵活可配, 大大简化使用接口开发的复杂性; 引入数据管理公共核心设施, 使得在实现业务逻辑时不用分心于繁琐的数据库方面的逻辑, 同时提供更加合理的模块划分的方法.

关键词: 应用框架; intelliWeb; 业务管理; 数据管理; 公共核心设施

中图分类号: TP 311. 11

文献标识码: A

目前, 已有了相对成熟的 J2EE(Java 2 Platform Enterprise Edition) 开发模式或开发框架. 但是, 这些模式或框架大多是针对传统的 BS(Browser/Server) 应用设计的, 存在界面表现能力差、操作方便性差(刷新、跳转)、响应速度慢、交互性差等问题. 随着企业级应用系统的逻辑越来越复杂, 版本更新周期越来越短, 需求变动和增长越来越快^[1]. 同时, 开发效率和开发难度始终也是围绕 Web 应用的开发的一道难题^[2]. 在 Web 层引入 intelliWeb, 该 Web 层中间件产品可以很好地解决这些问题并实现快速开发, 提高开发效率. 在引入 intelliWeb 的过程中, 开发人员并不希望自己的业务逻辑层的实现代码与之发生耦合. 因此, 往往需要在原有的核心代码与第 3 方产品之间建立一个胶合层, 利用这层代码完成对第 3 方产品的集成工作. 这里的胶合层应该是轻量的、灵活可配的. 本文设计并实现了一种基于 intelliWeb 的应用框架 iBAF.

1 iBAF 框架的基本结构

1.1 iBAF 框架

iBAF 框架依据 J2EE 的 3 层结构, 分别在各层中引入 intelliWeb、业务管理公共核心设施和数据管理公共核心设施, 基本结构如图 1 所示. intelliWeb 是架构在 J2EE 之上的应用快速开发平台, 由运行环境、运行框架(intelliWeb Runtime Framework)、集成开发工具(intelliWeb Studio)、应用框架集及业务应用组成. 其产品架构图如图 2 所示. 其中, 应用框架集由多个应用框架组成, 包括 intelliWeb 默认应用框架、intelliFlow 工作流应用框架, iBAF 应用框架.

iBAF 应用框架在 Web 层引入 intelliWeb 的 Web 层框架. intelliWeb 的 Web 层框架本身已经很成熟, 并且基于 Ajax 和 Rich Client 技术, 能实现局部刷新, 增强用户体验. 同时, intelliWeb 还提供丰富的插件和 Web 层的辅助开发工具, 大大加快开发速度. 这些都是当前已有的 Web 层框架无法比拟的. 但是, intelliWeb 平台的业务逻辑层和持久层存在扩展性不佳, 层次结合太紧密等问题.

为此, iBAF 框架分别在业务逻辑层和持久层引入业务管理公共核心设施^[3]和数据管理公共核心设施. 这两个设施是在总结当前流行的业务逻辑层和持久层框架基础上, 依据实际的项目需求和经验提出

收稿日期: 2008-08-13

通信作者: 余金山(1952-), 男, 教授, 主要从事软件工程、网络计算及人工智能应用的研究. E-mail: yjs@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(A0810013)

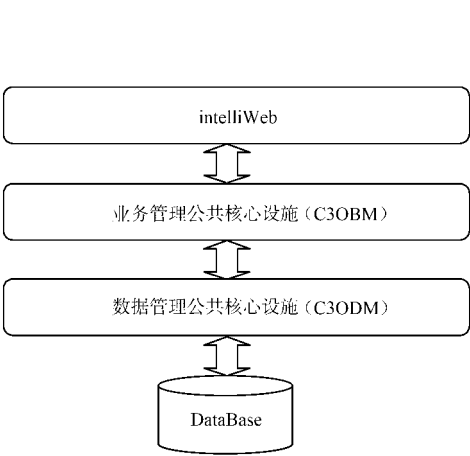


图 1 iBAF 基本结构

Fig.1 The basil structure of iBAF

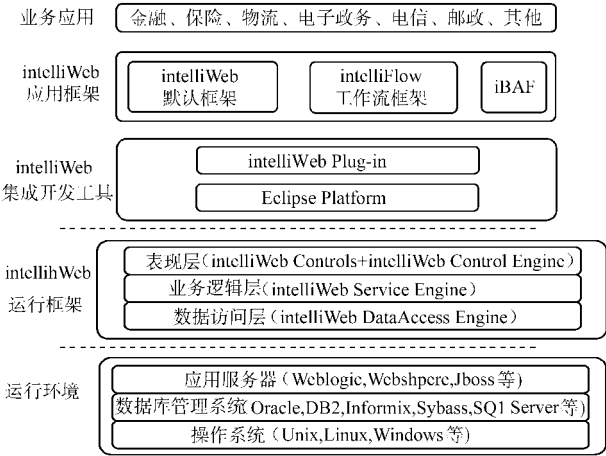


图 2 intelliWeb 产品架构图

Fig.2 The product architecture of intelliWeb

来的. 它使得各层结构划分更加清晰, 具有更强的可扩展性和实用性, 能够更好的满足实际的项目需求. 这样, iBAF 框架在具有快速开发这一属性的同时, 又增强了其扩展性, 层次清楚, 遵循当前的 J2EE 开发规范. 因此, 使得二次开发人员能够实现快速开发, 缩短开发周期, 又能够依据项目需求对框架进行必要的扩展, 从而更好的满足项目开发需要.

1.2 框架系统功能

本框架主要有以下 6 个功能: (1) 提供统一开发框架; (2) 基于角色权限管理的安全访问控制; (3) 系统登录统一认证模块; (4) 统一日志管理; (5) 消息文件、属性配置文件管理; (6) 统一权限认证模块.

选择了一个应用框架就选择了一个统一的开发框架, 开发人员可以基于此应用框架作二次开发, 用来满足一些特定的需求. 基于角色权限管理的安全访问控制为系统提供完善的安全与权限机制, 能根据用户的角色判断用户有没有权限执行特定的操作. 譬如, 只有拥有指定权限的人, 才能调用某个 URL 或者执行某个 Action 中的事件; 能根据用户的权限显示页面的某部分区域; 能根据用户的权限显示页面的某些按钮、某些控件或某些页面元素. 系统登录统一认证模块根据用户给出的信息(一般用户名和密码), 确认用户的身份是否合法, 认证过程还包括登录策略和用户授权信息初始化. 统一权限认证模块则在用户通过身份认证以后, 确认用户是否有权访问被保护资源. 统一日志管理为日志的管理者提供日志的基本信息和详细信息, 系统可以判断并通知管理者日志是否被更改过, 管理者可以根据日志当时的状况进行日志信息恢复的操作. 消息文件、属性配置文件管理能够修改并更新消息和属性配置文件.

2 iBAF 的设计与实现

2.1 iBAF 框架系统总体结构设计

在表现层引入 intelliWeb, Business Logic 代表开发商和开发人员现有的业务逻辑代码, 而 intelliWeb 包含了服务端引擎和客户端引擎两部分. 这两个部分是密切相关, 相互配合的, 他们共同完成了表现层的功能. 与业务逻辑的对接将主要通过 intelliWeb Server Engine 之间的交互来完成. 框架的业务管理公共核心设施和数据管理公共核心设施分别是采用 Spring 和 Hibernate 来实现的, 总体设计实现图如图 3 所示.

intelliWeb Server Engine 与 Business Logic 直接通过业务管理公共核心设施来起到承上启下的作用, 而在 intelliWeb 的 Server Engine 和 Client Engine 之间也有一些变化, 那就是控制层的引入.

对于 AJAX 应用而言, 在操作的过程中客户端存在着大量的远程过程调用(RPC). 这里的 RPC 并

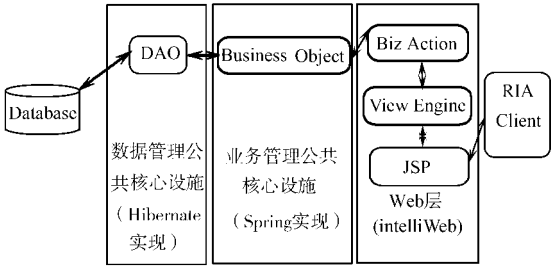


图 3 总体设计实现图

Fig.3 The general design structure

不是通常意义上所说的 Server-Server 的远程调用, 而是指从浏览器客户端到服务端, 利用 JavaScript 中的 iFrame 或 XMLHttpRequest 完成的方法调用. 这些来自客户端的 RPC 请求, 需要类似 Servlet 这样的对象来接收和处理. 在图 3 中可以看出, 在实现上, C3OBM 是利用了 Spring 来处理这些关于业务逻辑的请求, C3ODM 利用 Hibernate 处理数据的持久化.

使用这种方式主要基于如下 3 个方面的考虑. (1) 这些应用非常广泛的产品, 开发人员在原有的开发框架中可能已经使用了其中的某一个, 因此, 没有必要为业务逻辑重新修改业务逻辑过程. (2) 目前的技术人员可能已经熟悉其中的某种框架, 这有利于他们更快地了解本应用框架运转机制. (3) 开源框架往往都提供了很完善的配置方案和扩展机制, 例如其中的 Spring 就能很好地利用 AOP/ IOC 机制来辅助功能扩展.

在集成 Business Logic, Spring, intelliWeb, Hibernate 的过程中, 需要一些代码和配置来完成整个工作, 本应用框架可以来填补这些空缺.

2.2 iBAF 框架系统的实现

框架提供了所有应用期望的默认行为的 5 个类集合及一些基础功能模块. 这 5 个类集合分别对应安全权限控制包 (Security)、异常包 (Exception)、基础包 (Base)、公共服务包 (Common) 和工具包 (Util) 5 个类包, 而基础功能模块在应用业务类包 (Webapp) 中实现. 包的总体结构如图 4 所示.

2.2.1 基础包 包里是应用框架的所有基础类, 提供基础的日志、Action 和数据访问等服务. 其中的两个关键类:

(1) BaseAction 类. 应用框架的基础类 BaseAction 扩展了 AbstractBaseAction 类, 应用开发的全部 Action 必须扩展 BaseAction 类. BaseAction 增加了对所有 UI 控件的权限控制功能和身份认证

功能. 如果二次开发的应用 UI 控件, 需要受权限访问控制, 必须继承 BaseAction. 基于本框架的开发基础类都要继承 BaseAction 类. (2) 抽象 BaseService 类. 框架基础 BaseService 类, 所有业务逻辑 Service 类框架均继承该类, 可以根据实际业务需要扩展应用. BaseService 的主要功能: 目前实现的功能有 executeHQLDao 这个基础数据访问 DAO (根据实际需要可扩充); CheckInDB (String sql) 检查指定的字段的值在 DB 中是否已存在.

2.2.2 安全权限控制包 包里是所有和安全权限控制相关的类, 提供密码加密、安全过滤和保存登录用户信息等服务. 其中的两个关键类: (1) 安全过滤类 (SecurityFilter). 必须放在 Web.xml 的 filter 中声明. 此类的主要功能: 过滤没有授权的 URI 请求; 保护受安全条件保护的 Web 资源. 当用户在未登录的情况下登录系统的时候, SecurityFilter 过滤器会检测用户的系统登录情况; 用户没有登录的情况下, 访问应用系统会自动把页面跳转到系统登录界面. (2) 登录用户基础权限信息类 (SessionObject). 保存了用户登录的信息, 同时保存登录后用户的身份信息等到 Session 中.

2.2.3 公共服务包 包里是系统公共核心类, 为系统提供常量服务、消息服务、配置文件服务及分页服务等. 其中的关键类: (1) Config 类. 即读取 ApplicationConfig.properties 配置文件信息. (2) Constants 类. 即常量文件类. (3) Message 类. 即读取 ApplicationMessage.properties 自定义信息类.

2.2.4 工具包 包里是应用框架的所有工具类, 主要为系统提供 intelliWeb 与 VO 类型转化、文件、算法、唯一键值、字符和日期工具等服务. 其中的几个关键类: (1) DateUtil 日期工具类和 UniqueKeyUtil 产生唯一 Key 类. DateUtil 开发接口, 主要是针对重复使用的 DateTime 日期时间类; 而 UniqueKeyUtil 开发接口, 产生唯一 Key 值类. (2) StringUtil 字符串截取工具类. StringUtil 开发接口, 主要是针对字符串的操作类, 譬如去掉空格、截取部分字符、替换等等操作. (3) BeanUtil 转换工具类. BeanUtil 开发接口, 主要是 DataSet 与 VO, VO 与 DataSet 互相转换的接口类. 开发接口主要有 RecordToVO (把

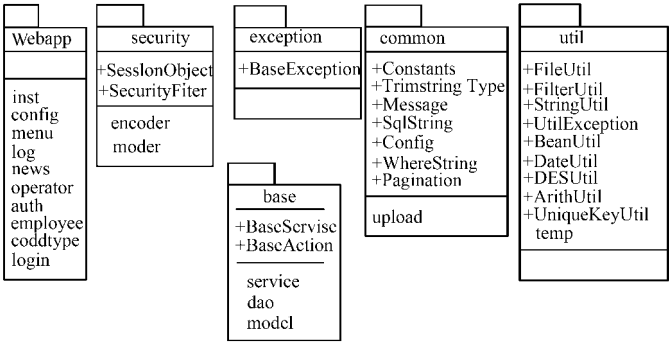


图 4 总体包结构

Fig. 4 The general package structure

Dataset 转为单个 VO 对象), toSingleVO(把 Dataset 转为单个的 CustomDataset 对象), ListToDataset (把集合对象转为 CustomDataset 对象) 等等接口.

2.2.5 异常包 包中只有一个 BusinessException 类, 是所有异常类的父类. 所有异常类都要继承此类, 并对此类进行扩充实现对异常的处理.

2.2.6 应用业务类包 实现了部分常用的应用, 包括权限管理、登录管理、日志管理、员工管理、机构管理、菜单管理和角色管理等应用开发中的常用模块, 并对应到 Auth, Login, Log, Employee, Inst, Menu 和 Operator 等包来实现. 同时, 它们严格按照 3 层结构, 区分出 Web 层、业务逻辑层和持久层, 从上到下实现松耦合.

应用框架能够为基于 Spring, Hibernate 等开源框架的开发模式, 提供与 intelliWeb 集成的范例性实现代码. 同时, 为基于此框架的大型应用软件系统提供所有应用期望的默认行为的类集合, 以及一些基础功能模块. 具体的应用通过重写子类(该子类属于框架的默认行为)或组装对象来支持专用的行为, 从而提高软件的代码和设计重用性^[4]和系统的可扩展性, 缩短大型应用软件系统的开发周期, 提高开发质量, 并且可以增强大型应用软件的可扩充性和代码重用性^[5].

3 结束语

设计并实现的基于 intelliWeb、业务管理公共核心设施和数据管理公共核心设施的应用框架. 经过一些实际项目的使用和评估, 并在此基础上进行了一些改进. 目前, 框架稳定、性能优越、可扩展性强, 表现出相当高的实用价值和推广价值.

参考文献:

[1] 邓虎成. 基于 J2EE 的应用框架的设计与研究[D]. 成都: 西南交通大学, 2005.
[2] 焦治平. J2EE Web 层应用框架研究[D]. 西安: 西安电子科技大学, 2005.
[3] WALLS C, BREIDENBACH R.Spring in action: 中文版[M]. 李 磊, 等译. 北京: 人民邮电出版社, 2006.
[4] 陈传波, 潘大为. 可扩展的企业应用框架[J]. 计算机工程, 2003, 29(9): 53- 57 .
[5] 李英军, 吕 建, 王宏琳. 面向对象应用框架在油气勘探领域的应用研究[J]. 软件学报, 1999, 10(4): 349- 354 .

The Design and Implementation of an Application Framework iBAF Based on intelliWeb

LU Xiao-xing, YU Jin-shan

(College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China)

Abstract: In order to reduce the design complexity of J2EE and cut down the time of software development, the iBAF application framework based on intelliWeb was designed and implemented. This framework which imports intelliWeb, common core component of business management and common core component of data management in Web layer, business logic layer and data persistence layer respectively provides some stable and basic services and modules for secondary development. These also show the essentiality and superiority of the framework in development of projects. In the course of cementing layer between the original core codes and third party product realized, common core component of business management is imported to make cementing layer more light weight, and flexible and simplify the interface development, and the common core component of data management introduced making the implementation of the business logic in which no distract to tedious logic of database and to provide the more reasonable way on the modules partition.

Keywords: application framework; intelliWeb; business management; data management; common core component

(责任编辑: 陈志贤 英文审校: 吴逢铁)