

构建工作流驱动的符合 J2EE 标准的 广东省民政业务支持系统^{*}

倪 健

(中山大学软件研究所, 广东 广州 510275)

摘 要: 叙述了用工作流和 JAVA 技术开发广东省民政业务统一软件的架构设计思想, 开发出一个流程驱动的主要基于配置开发的民政业务支持系统 (BOS)。

关键词: J2EE; 工作流; O-R Mapping; BOS; QL

中图分类号: TP311.52 **文献标识码:** A **文章编号:** 0529-6579 (2004) S1-0132-03

通过对广东省民政业务的调研, 我们发现业务主要是基于各种信息表格的 CRUD 操作, 种类繁多, 且具有典型的流程性。开发目标是一个基于 J2EE 的 B/S 结构的应用系统。为此我们做如下架构开发, 即一个流程驱动的主要基于配置开发的业务支持系统 (BOS)。流程驱动是指需要使用工作流引擎来驱动流程。以下是架构设计的几个部分。

1 前台框架的构建

目前的 j2ee 相关的 web 前台开发框架主要由 MVC1/MVC2 的两种模式构成。本系统选择最简单的以 servlet/jsp 为核心的免费的成熟前台框架, 不使用 XML 等技术, 也不考虑和 EJB 的集成。

1.1 系统的前台 (表现层) 框架主要使用目前流行的 struts 1.1 来进行开发

struts 目前是使用最为广泛的 MVC 2 的前台框架, 事实已上成为前台开发的默认标准, 但开发工作量稍大, 不适合做快速开发, 相对于最简单的 jsp+bean 的模式, 它在表现/逻辑的分离带来了可维护性的提高。但是因为需要明确划分各种层面, 开发的工作量相对来说有所增加。另外, 配置文件维护较困难, 且不适合交互程度要求比较高的业务。

尽管如此, 我们可以对它存在的缺陷做一些修订, 例如尽可能使用封装好的架构来开发, 如大部分工作将使用封装好的 dbforms 通过配置做开发, 使用 easy struts/struts console 等集成度较高的工具来维护相关配置文件。

1.2 对于交互度要求比较高的模块, 如查询定义, 报表定义, 使用 SOFIA 进行开发

SOFIA 是基于事件驱动的模式开发, 适合开发交互程度比较高的业务, 又有较丰富的界面组件, 适合做快速的数据驱动类型业务开发, 和 Eclipse / Dreamweaver 等工具有较高的集成度, 开发的代码行数较少。它完全基于 servlet/jsp 技术, 在主流应用服务器上都可以成功部署, 并且不依赖于 EJB 等较复杂的技术, 可移植性好, 性能也比较好。

但由于各层间捆绑较死, 不易扩展和封装, 不容易和工作流以及相关的 O-R Mapping 工具集成。

使用 SOFIA 来开发查询/报表定义模块比较适合, 能减轻一部分开发工作量, 即便存在风险, 也不会对主要业务有影响。

1.3 struts 与 sofia 不会互相影响

使用 struts 1.1 来做主要业务的前台框架而 sofia2.0 来做辅助业务的开发框架 (包括前台框架), 这两部分模块互相独立, 不存在互相调用的问题。

2 业务逻辑层和数据访问框架的构建

民政项目的业务需求缺乏复杂逻辑, 只有少部分涉及较复杂的业务逻辑, 简单的 java 类也可以达到同样的效果, 不使用 EJB, 只用普通的 JAVA 类来封装。

民政业务对数据访问提出了较高的要求。要求应用系统同时独立于应用服务器和数据库系统, 对设计数据访问框架有了约束性的要求。

通过分析也发现, 民政业务基本上不用考虑复杂的事务, 尤其是分布式事务, 数据访问的中心应

^{*} 收稿日期: 2004-04-04

该集中在数据的操作上。

目前, 对数据访问方式的基本做法有:

(1) JDBC 直接访问: 缺点是工作量大, 不容易移植, 质量也比较难保证。

(2) EJB 方式: 主要有 2 种方式, Session Bean 和 Entity Bean。在 session bean 里面使用 Jdbc 来访问数据库, 缺点和直接使用 Jdbc 类似, 开发工作量还有所增加。

经验证明, EJB 是很难在不同的应用服务器和数据库间移植的, 由于民政业务不用考虑事务, 尤其是分布式的事务, 用 EJB 没有优势。

(3) JDO 方式: 相对于 EJB 是一种轻量级的解决方案, 规范还不够成熟, 前景不太明朗, 对 QL 的支持还比较有限, 开发受到一定限制。

(4) Hibernate 是 Sourceforge 提供的一个开源的 O/R Mapping 框架: 对 QL 支持比其他产品好, 使用 QL 可以很好的做到数据库无关的开发, 对所有主流的商业数据库和开源数据库都有支持。

经过分析, 系统架构中采用 Hibernate 数据访问的基本结构使用 DAO/Value Object 的模式对 Hibernate 的进行封装, 开发人员只需要学习 QL 就可以完成数据的访问操作, 对基本单表的 CRUD 操作, 也提供了缺省实现, 进一步简化了开发。

大部分业务由 DBForm 封装的数据访问部件完成, 这部分是直接基于 JDBC 的, 也可用连接池。由于 Dbform 是可配置的, 由配置文件来定义数据操作, 修改比较简单。

由 Dbform 处理的业务基本都是 CRUD 操作, 不会使用特殊的 SQL 和统计函数, 只要仔细处理日期和二进制对象, 就可以做到跨数据库操作。且 SOFIA 的 datastore 本身是跨数据库的, 系统的其他业务, 特别是涉及到统计和计算的部分由封装好的 Hibernate DAO 来完成, 可以做到与数据库无关, 开发工作量也很小, 只需定义 QL 即可。

3 工作流引擎和业条模块

要构建一个流程驱动的系统必须有一个工作流

引擎做核心, 本次开发的做法是使用开源的工作流引擎做 2 次开发的。架构最终还是选择使用 OBE 作为系统工作流引擎的实现, 原因是它符合 WFMC 规范, 出现问题的时候便于切换到其他开源或商用引擎。通过对 OBE 的封装, 将工作流引擎集成进了架构。

业务需求和工作流相关的另外一个难点是远程工作流, 因为系统中有两级业务系统, 审批需要跨服务器进行, 必须使用远程工作流 (也就是工作流引擎之间的交互)。

根据经验, 业务系统中 80% 的业务是对表单的操作, 而这种类型的业务具有相当的共同点, 各个处理的差异只是数据的差异, 可以通过使用公用的模块, 加上配置文件来生成业务系统, 避免重复开发。我们选择了开源的 Dbforms 进行改造来达到此目的。

Dbforms 是一个开源的 jsp 数据访问应用, 符合 j2ee 标准, 便于移植, 它提供对数据表的基本操作的全部封装, 在此基础上便于修改集成。还提供了在更新数据后自动调用的接口, 可以利用这些接口, 在完成数据更新操作以后, 调用工作流的接口更新状态, 实现和工作流系统的无缝集成。

可配置业务的开发工作主要是对相关的 Dbform 配置文件和数据校验定义文件的编写和维护。

不可配置的业务开发则需要完整地开发相关的 action/jsp 和数据访问对象。在特定的数据操作更新以后, 要调用架构封装好的 OBE 接口来更新流程状态。

4 系统的总体结构类图及存在的问题

系统的总体结构类图见图 1。

系统存在的问题主要是多级工作流驱动的问题, 由于开源的工作流产品没有这方面的支持, 所以导致一些实施上的限制, 例如同一个地级市不能部署多于一个数据库。

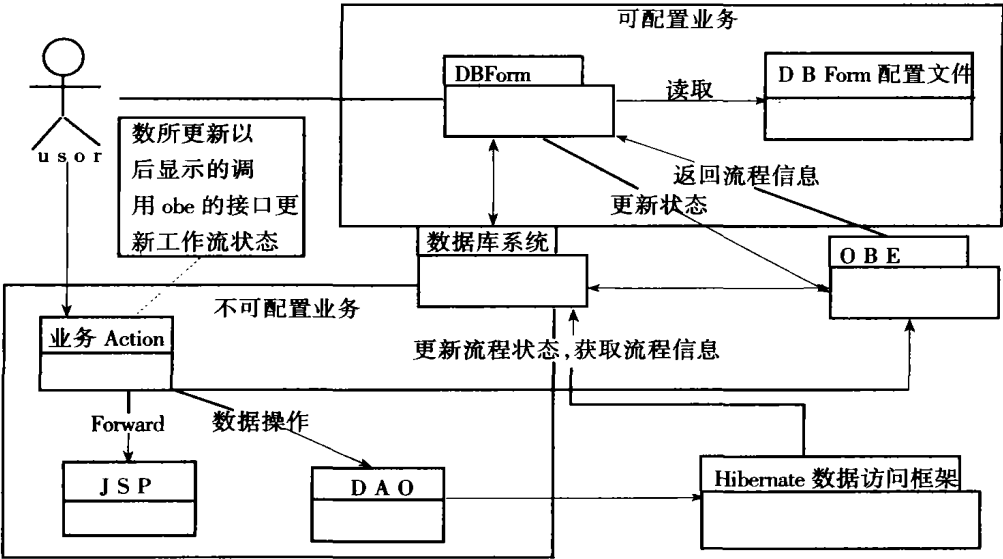


图 1 系统总体结构类图

Fig. 1 Object architecture

参考文献:

[1] WORKFLOW HANDBOOK, Lighthouse Point, FL, 2004.

[2] [美] 斯麦里. JAVA 程序设计教程[M] . 北京: 电子出版社, 2002.

[3] DAO OF WEB DESIGN John allsopp. Navneet, 2003.

Design a Workflow-driven J2EE BOS for the Civil Affair Departments in Guangdong Province

NI Jian

(Software Research Institute, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: To satisfy the different demands of the Departments of Civil Affairs from different areas, a BOS using workflow and J2EE technology is built by using our beveloped software. The highlight of this BOS is using open source tools of struts, sofia and Dbform.

Key words: J2EE; workflow; O R mapping; QL; BOS