

基于智能表单的事务处理系统

阳奇, 黄帆, 林镇灿, 庞国明

(华南理工大学 计算机软件学院, 广东 广州 510006)

摘要: 采用智能表单作为信息载体, 结合 workflow 技术, 提出了一种轻量灵活的事务处理系统。根据实际业务处理过程中半结构化以及规则多样化的特点, 系统中的信息多以半结构化数据进行存储, 可以方便地定义事务类型, 并加入一个检索模块提供对业务信息的高效全文检索服务。

关键词: 智能表单; 半结构化数据; 事务处理; 工作流

中图分类号: TP391.4

文献标识码: A

Transaction processing system based on intelligent form

YANG Qi, HUANG Fan, LIN Zhen Can, PANG Guo Ming

(School of Software Engineering, South China University of Technology, Guangzhou 510006, China)

Abstract: Based on the combination of workflow engine and intelligent form, which is used as information carrier, this paper provides a lightweight and flexible transaction processing system. According to the semi-structured and diversified features in actual business transactions, the information in the system is mainly stored as semi-structured datum, which can define the category of the service type easily. The system also has a search module which can provide efficient full-text retrieval services for business information.

Key words: intelligent form; semi-structured data; transaction processing; workflow

信息时代, 及时获取以及准确处理业务信息是政府、企业等部门在日常办公中所要面对的重要工作。以纸张为信息载体的工作程序, 无论是在效率上、准确度上以及存储上都难以满足实际业务需要, 这使得业务管理人员必须引入计算机技术的支持, 由此一些利用计算机, 按某种预定规则自动传递文档、信息或者任务的系统出现了。这些系统很多以智能化的表单取代了纸张成为传递信息的载体, 并为事务处理活动建立了计算机模型。随着技术的进一步发展, 以人为中心的业务流程管理BPM(Business Performance Management)与以管理系统为中心的工作流(Work Flow)被人地区别开来^[1], 后者主要从技术角度出发成为了计算机学科研究的重点。

目前很多企业都推出了各自的智能表单系统, 如书生的SureForm、微软的InfoPath等, 这些系统很多除了系统自身十分庞大外, 有的还要特定的后台服务器支持, 难以与中小型网络服务相结合^[2]。本文通过分析在政府、企业当中行政与商业事务处理的特点, 提出了一种可用J2EE轻量级Spring框架实现的, 以智能表单为核心结合工作流技术

的事务处理模型, 这种模型具有轻量、灵活、易于与B/S网络服务相结合的特点。

1 模型的建立

为了将实际的业务处理过程转换为计算机模型, 并在此基础上实现数据的传递以及业务流程的管理, 系统必须要建立一个元模型。这个元模型将由一些特定的元素所构成, 业务处理的过程通过这些元素的信息交互来实现。如表1所示系统模型中主要定义了活动、表单模块、表单、用户、角色等元素。

表1 事务处理模型中的元素

元素名称	相关概念
活动	事务处理流程中的一个步骤
表单模板	用于存储活动的相关操作以及显示界面
表单	存储了业务流程的路由相关信息以及处理的业务内容
用户	系统中的用户
角色	与一个具体活动或操作相关联的组织实体

系统首先要制定一个事务处理流程, 将流程路由步骤信息存储在表单当中, 并且定义流程所需的操作以及显示

界面存储在表单模板当中。一个具体的事物处理流程包含一个表单元素以及一个表单模板元素，而在整个系统当中表单模板与表单具有一对多的关系。 workflow引擎在运行一个事务处理流程的实例之前，系统必须初始化与这个事务处理流程相关的角色，规定好角色所拥有的操作，以及将这个角色与系统中的用户加以绑定。这样角色就成为了用户与表单模板、表单等模型中其他元素相关联的纽带。

从面向对象观点看，事务处理过程实例是 workflow模型类的一个具体对象。 workflow在过程实例的执行过程中， workflow引擎将生成有关的活动实例并根据模型中的控制规则协调这些活动实例之间的顺序关系，同时根据数据流动关系的定义完成活动实例之间的数据传送^[3]。

2 系统总体结构

事务处理系统采用J2EE中的Spring框架，并使用Spring的MVC(Model-View-Controller)作为系统网络访问层的界面与控制逻辑架构。Spring框架通过采用依赖注入DI(Dependency Injection)机制实现了控制反转IOC(Inversion of Control)技术，通过采用DI技术使Spring成为一种轻量级，所实现系统内部各模块耦合性低，易于测试的框架^[4]。整个事务处理系统是传统的B/S网络服务三层架构，事务处理系统整体结构如图1所示。

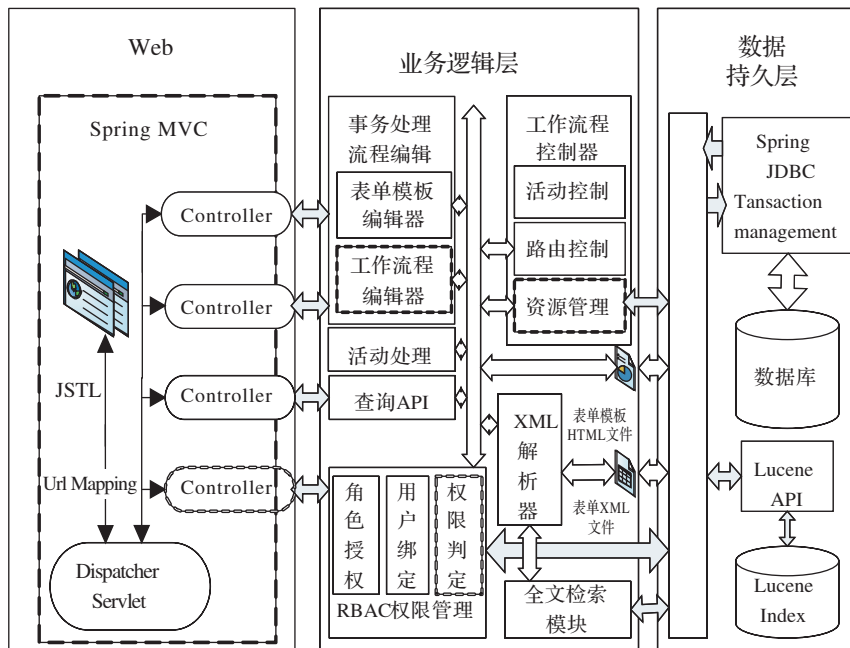


图1 事务处理系统整体结构图

(1)在数据持久层，系统采用了Hibernate对象关系映射框架。Hibernate可以方便地与Spring框架中的JDBC事务管理机制相结合。数据持久层当中除了使用传统的Hibernate关系映射框架外还采用了由Hibernate提供的Hibernate Search^[5]技术。通过Hibernate Search与Lucene^[6]相结合，可以提供对数据库中数据高效的全文检索服务。

(2)业务逻辑层包含了与智能表单和 workflow相关联的一

系列接口与实现。其中有进行表单模板与流程编辑的事务流程编辑器，采用RBAC模型进行访问控制的权限管理模块以及进行流程路由与活动控制的工作流程控制器等业务逻辑单元。这些业务逻辑单元提供了系统表示层和数据持久层的联系。

(3)Web层向用户提供了应用服务的图形界面。这一层采用了Spring MVC框架，JSTL(JSP Standard Tag Library)用于实现MVC中的View页面。JSTL依靠其标准库所提供的标签保证了其所实现页面的高可靠性与可移植性，能让程序开发人员与界面美工人员进行更良好的合作。

3 主要技术细节

事物处理系统的运作是通过组成系统的各个模块以及与 workflow相关的各个元素相互作用的结果，其具体实现主要包括以下技术细节。

3.1 事物流程编辑

系统中的事务处理流程必须要创建后才能运行。一个具体事务处理流程的创建主要包括两方面，一是要设置好 workflow的基本路由路径，另一方面是要编辑路径中各个具体活动的界面与操作。事务处理流程的创建方式主要有采用并修改原有模型以及完全重新创建两种，其创建工作主要由以下两个模块完成。

(1)表单模板编辑器：主要用于编辑流程中具体活动的操作界面。表单模板是一个HTML格式的文件，事物处理系统中定义表单模板 $FT = \{a_1, a_2, \dots, a_n \mid n = 1, 2, \dots\}$ ，其中 a 为表单模板划分成的表单模板域，单个表单模板域中包含了一些文本信息以及如input、textarea等表单项。通过表单模板编辑器，用户可以采用拖拉控件的方式编辑出流程中各个活动所需的表单模板域，域中的表单项以及相关操作。

(2) workflow编辑器：系统中一个事务处理流程的路由信息要通过 workflow编辑器编辑得到。 workflow编辑器可以帮助用户定义事物处理流程中的各个活动，活动的执行顺序以及这些活动所涉及到的角色、表单模板域等相关资源。 workflow编辑器通过调用XML解析器的相关接口将编辑得到的流程路由信息以XML格式写入与当前事物相对应的表单中。根据实际业务处理过程半结构化的

特点，系统中XML解析器通过采用JAVA的DOM(Document Object Model)方式结合正则表达式来提供更改表单内容的方法。

通过以上创建得到的只是一个事物处理流程的模型而非运行的实例，模型可以通过修改得到一个新的模型。直到用户点击执行时，系统才真正创建了 workflow模型类的一个具体对象。

3.2 角色访问控制

访问控制是信息系统安全的一个重要方面，更是事物处理系统进行工作流程控制的基础。事务处理系统采用了基于角色的访问控制RBAC(Role-Based Access Control)模型，这种访问控制模型是当前主流的访问控制模型，其最早由美国国家标准与技术研究所(NIST)提出并定义^[7]。系统为了让事物处理的过程更为灵活，对RBAC模型进行了树形扩展加入受限继承，使之成为以树形事项为作用域的RBAC模型。

在访问控制模型中，本文第一小节中所提到的用户元素集合被定义为 U ，角色元素集合被定义为 R 。除此之外，资源集合为 D ，操作集合为 O ，这里所谓的资源与操作主要是指表单和表单模板以及与之相对应的一系列操作，当然也包括系统中的其他一些资源及其操作。定义树形节点集合 N ， $E \subseteq N \times N$ 为树形结构中的一条边，而小于 N 和大于 E 的部分就是一棵角色访问控制树。以上定义的元素与节点之间的谓词关系包括： $P \subseteq D \times O$ 为操作关系； $\alpha \subseteq P \times R$ 为授权关系； $\beta \subseteq U \times R$ 为隶属关系；角色和节点对应关系 $RN \subseteq R \times N$ ；资源和节点对应关系 $DN \subseteq D \times N$ ；操作和节点对应关系 $ON \subseteq O \times N$ ；权限和节点对应关系 $PN \subseteq P \times N$ ；授权和节点对应关系 $\alpha N \subseteq \alpha \times N$ ；隶属和节点对应关系 $\beta \subseteq \beta \times N$ ；角色、操作、授权和隶属与节点的对应关系 $@ \subseteq (R \cup P \cup \alpha \cup \beta) \times N = RN \cup PN \cup \alpha N \cup \beta N$ 。权限控制树中的一个节点 n 上的角色 r 和权限 p 分别被记为 $r(n)$ 与 $p(n)$ 。

定义1：角色 r 在节点 n 上的权限集 $privileges(r, n)$ 是一个函数 $RN \rightarrow 2^P$ 。

定义2：角色 r 在节点 n 上的用户集 $users(r, n)$ 是一个函数 $RN \rightarrow 2^U$ 。

在节点 $n \in N$ 上，用户 $u \in U$ 隶属于角色 $r \in R$ ，记为 $u\beta r@n$ ， $u\beta r@n \Leftrightarrow \langle u, r \rangle \in \beta$ 且 $\langle u, r \rangle @n$ ；在节点 $n \in N$ 上，权限 $p \in P$ 被授予角色 $r \in R$ ，记为 $p\alpha r@n$

$p\alpha r@n \Leftrightarrow \langle p, r \rangle \in \alpha$ 且 $\langle p, r \rangle @n$ 。由以上，系统的访问控制实现如图2所示。

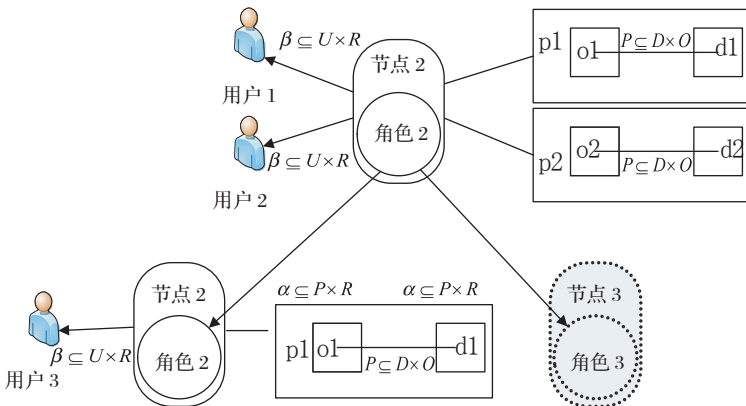


图2 RBAC树形模型

通过以上元素以及元素之间的关系，角色访问控制模

块便为实现事务流程的调度提供了条件。

3.3 流程的调度

事物处理流程的调度由工作流程控制器负责。通过查看表单中的路由信息得到下一步的活动后，工作流程控制器会依据表单中的设置将下一步活动所要用的表单模板域等资源以及相关操作权限授予用户。一旦活动的相关授权完成，流程控制器便开始等待获授权用户的操作。直到用户完成了此项活动的操作并提交，活动处理模块将业务信息写入了表单当中，工作流程控制器才会在表单中将当前活动步骤的状态字段置为“完成”，同时剥夺当前用户的权限并查看下一步活动。工作流程控制器便是通过重复上述操作来完成整个事物处理流程的。

以上可以得出，系统对事物处理流程的调度是通过用户权限的授予与剥夺来实现的。由于采用了树形扩展的权限访问控制方式，获得授权的用户在活动处理过程中，根据权限，既能自行处理也可以选择授权其他用户处理以提高事物处理过程的效率以及灵活度。

3.4 业务信息检索

在事物处理流程的执行过程中以及整个流程执行完成后，流程的表单都会记录下当前流程的业务信息。表单记录的这些业务信息不光能为用户的业务处理活动提供参考，而且作为档案更能够指导用户未来的工作。

表单是采用XML格式的半结构化数据存储的，由参考文献[8]中得出传统的查询方式主要通过遍历整个XML文件实现，而这种查询方式效率不高。为了实现对表单这种半结构化数据的高效全文检索，系统采用了Hibernate Search技术结合上Lucene作为检索引擎。Hibernate Search提供了可供扩展的接口，利用接口编写如PDFHandlerBridge、ExcelHandlerBridge等处理桥提取文件中有用的文本信息，便可以实现对PDF、Excel等格式文件的全文检索。同样针对表单，系统也提供了FormHandlerBridge处理桥，以Hibernate POJO持久类Form(表单)为例，配置代码如下：

```
@Entity
@Indexed
public class Form {
@Id
@DocumentId
private Integer id;
...
@Field(index=Index.TOKENIZED, store=Store.NO)
@FieldBridge(impl=org.formsystem.bridge.
FormHandlerBridge.class)
private String content;
...
}
```

通过采用以上配置，系统便能在表单创建以及遭到更

(下转第73页)

(上接第69页)

通过采用以上配置,系统便能在表单创建以及遭到更改时提取其中的有用文本信息,并同步更新表单所对应的Lucene索引为检索提供条件。Hibernate Search技术的采用为系统提供了包括表单在内多种格式文本文件的数据库全文检索服务。

本文通过研究现有政府、企业当中行政与商业事务处理需要,提出了一种基于智能表单的事务处理系统,并结合了工作流,树形权限访问控制机制以及Hibernate Search等技术。在下一步的工作中,将改进系统中的业务流程调度,让系统在实际运作过程中变得更为灵活高效。

参考文献

- [1] PYKE J. Fore word. 2008 BPM & Workflow Handbook [M]. 2008.
- [2] 唐文忠,莫伟栋. 面向领域的模型驱动智能表单系统的框架设计[J]. 北京航空航天大学学报, 2007, 33(9).

- [3] 曾炜, 阎保平. 工作流模型研究综述[J]. 计算机应用研究, 2005, 33(9).
- [4] LADD S, DAVISON D. Expert Spring MVC and Web Flow [M]. 2006: 7-8.
- [5] Hibernate.org – Hibernate Search [Z]. <http://www.hibernate.org/410.html>.
- [6] HATCHER E, GOSPODNETIC O. Lucene in action [M]. Manning Publications Co. 2005: 42-43.
- [7] FERRAILOLO D, KUHN R. Role-based access controls[C] // Proc. of the 15th NIST-NCSC National Computer Security Conference. altimore, Maryland, USA: NIST-NCSC, 1992: 554-563.
- [8] 王真星, 顾宁, 施伯乐. 基于本体的半结构化数据的柔性查询[J]. 计算机研究与发展, 2003, 40(11).

(收稿日期: 2008-12-17)