

一种符合快速开发要求的 OA 系统框架的实现

查伟忠, 杨季文

(苏州大学 计算机科学与技术学院, 江苏 苏州 215006)

摘要:通过分析 OA 系统的内容及其处理特征, 结合具体的项目实施经验和成熟的软件设计方法, 提出了一种符合快速开发要求的 OA 系统的框架, 详细描述了该框架的设计思想, 简要说明了使用和扩展过程, 同时提出了从系统的整体角度综合设计系统安全性的方法。利用该框架具体部署一个 OA 系统时, 能获得项目初期的快速开发能力, 中期的灵活、快速部署能力, 后期的简便维护能力, 同时能获得较高的系统整体安全性和集成能力, 适用于高校、政府机关和中小企业的 OA 系统开发应用。

关键词:办公自动化系统; 开发模式; 快速开发; 多层结构; 角色控制

中图分类号: TP317.1

文献标识码: A

文章编号: 1673-629X(2008)01-0093-04

Implementation of a Framework for Rapidly Developing OA Application

ZHA Wei-zhong, YANG Ji-wen

(School of Computer Science and Engineering, Suzhou University, Suzhou 215006, China)

Abstract: Analyzing the characters of data and business logic in the OA application, taking the practice of deploying OA application and mature technology of software-architecture as the corner stone, goes into the designing process of a framework which shortens the design cycle of OA application. Briefly describes the way to put it in practice and make extended design, also makes some advices on how to design a secure application. With this framework, can provide an OA application with QS of rapid-develop, flexible-deploy, easy-maintain, security and integration, especially fits the requirement of high-school, government and enterprises.

Key words: OA application; design pattern; rapid-develop; n-tier; role-based access control

0 引言

OA、OA 系统等相关名词已流行了许多年, 各种相关的系统软件也不断地推陈出新, 此类应用系统的社会需求日益提高。目前 OA 系统已成为电子政务建设的最基础的应用和整合平台, 也是企业进行信息流管理的最直接手段。OA 的应用是政府和企业管理现代化的标志。

通过归纳和总结具体的 OA 项目实施经验, 提出了一种符合快速开发要求的 OA 系统框架原型, 图 1 是该框架的简要组成图。框架的基础结构设计遵循了传统的多层结构设计方式, 其中用户接口层完成用户视图的定义与控制, 应用逻辑层构建主要的框架功能, 并根据不同的功能采用不同的技术实现, 数据层完成数据存储定义和操作规则定义。整个框架设计体现了

基于定义和配置的设计理念, 以期达到快速开发、灵活部署、维护简便的特征。

1 OA 系统核心内容抽象

OA 系统的核心内容可以抽象为“文”, 一个“文”对应一类待处理问题的内容抽象, 对应为一类数据及处理方法, 如可以是一篇公文、一次会议安排、一次车辆调度, 或者是一份财务报告、一份合同书等。这些都是 OA 系统的“数据”, 伴随着处理这些数据, 系统需要一些必要的功能, 如流转、状态监控等。

“文”的设计是 OA 系统的核心内容, 框架设计中把一个“文”的设计分割为内容、格式、功能、控制设计等。这几部分的设计依据了 MVC 及其几种变体设计模式^[1,2]。

2 内容设计

内容设计通过配置“文”真实数据源来提高数据设计的灵活性和扩展性, 数据源可以是用户输入, 如起草一篇公文; 可以是来自于另一系统; 或者是两者的结

收稿日期: 2007-04-06

基金项目: 江苏省高校自然科学基金计划资助项目(04KJD520164)

作者简介: 查伟忠(1971-), 男, 江苏苏州人, 实验师, 硕士, 主要从事 WEB 技术和应用研究; 杨季文, 教授, 主要从事中文信息处理研究。

合。框架设计强调数据源的定义,具体内容由该“文”的配置文档定义。

当数据是本系统独立新生成的,数据源配置包含产生该数据的基本信息,如某个 Form 的基本信息(位置、名称等)或某个方法信息等,同时定义该类数据的存储信息,如对应数据库的基本信息(位置、名称等)。

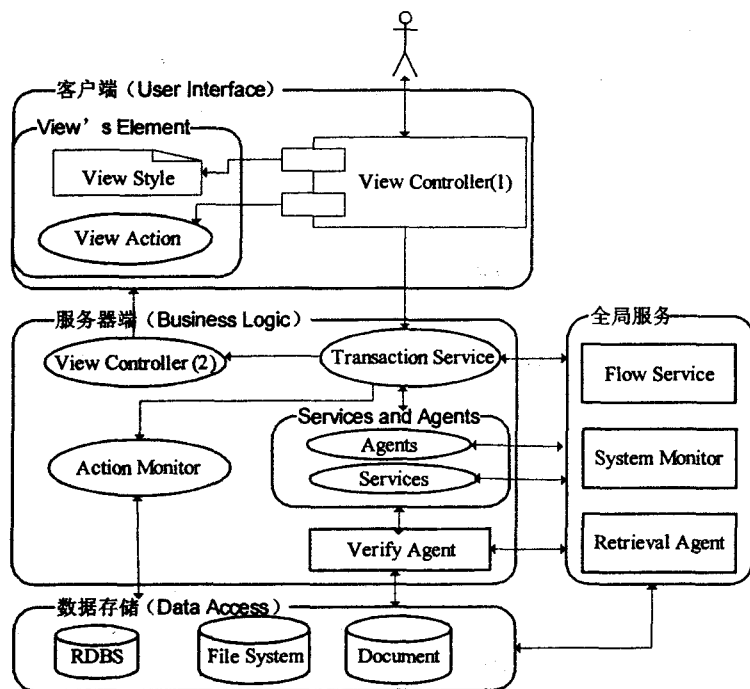


图 1 框架简要组成图

当数据来自于一个异构系统,需对方系统提供数据接口信息^[3],如 WEB Service^[4]的信息,包括返回数据的 Schema 等。因此数据源配置文档中应定义三部分信息,首先是调用对方系统的 WEB Service 的基本信息,其次是根据对方系统提供的 Schema 信息设计的 XSLT 转换定义(系统内部数据的结构定义),然后是一个方法信息和该类数据的存储信息。

数据的结构定义是一个独立的配置文档,如一个 XSLT 文档,框架设计中同时定义了此类数据的标准结构,由一个 XML 文档组成,主要是为了数据处理方法的标准化,如后期数据整理、检索等;方法是一个服务器方程序,如 agent 或 Service 等,负责数据转换和本系统存储等功能。

数据生成后,由于这些数据本身在 OA 系统内一般不经常作改变,框架设计时用一个文档来描述,并且进行存储,在实现时抽象为相应的对象和操作。

内容设计中的另一项任务是定义一个“文”在其生命周期中易变的部分,如处理过程、内容等,在框架设计中,把此部分内容绑定到了一个“文”的不同生命周期阶段的功能部分,进行单独处理,以提高整个框架的可配置性和灵活性^[5]。

3 格式设计

格式为“文”在系统中的视图定义,是纯的显示格式,对应为一个视图或不同状态下的多个视图,如起草格式、流转格式等。一种视图对应一个格式定义文档,格式定义文档可以是 C/S 模式中的一个 Form、Page 等,也可以是 B/S 模式中的一个 XSLT 文档等。

“文”内容的显示控制设计,根据配置可以采用先期生成或后期转换的方式。先期生成,是指在“文”的各个不同生命周期阶段,由状态控制程序把“文”的内容与对应格式合成一个独立物理文档,如 DOMINO 系统中的 document, PDF 文档或 HTML 文档等;后期转换是指在“文”需要显示时,由格式文档定义的初始化方法进行运行时转换,而不生成物理文档。一般可以结合使用,对于“文”生命周期中不易改变部分采用先期生成,而需经常改变部分则用后期转换方式,然后进行复合显示,以提高性能和显示效率。

4 功能设计

功能是指在一个“文”的不同生命周期阶段应具备的各种操作及辅助功能。框架设计中把功能区分为系统功能、共享功能和视图操作等部分独立设计。系统功能为具备一定独立性和整体性的功能模块,如“文”的流转等;共享功能是指相对独立,但与一个“文”的不同生命周期阶段有一致对应关系的,如监控、编号管理等;视图操作是指在某一特定视图上具备的独立功能模块,如保存等。框架设计中对三类功能设计采用了不同的处理方法,基本宗旨是使每一功能均可通过配置文档与具体视图的操作或事件关联,以达到项目初期能快速定制出符合需求报告的项目原型。本节余下内容通过描述流转、监控、保存功能的设计来阐述不同类型功能的处理方法。

1) 流转设计。

在 OA 系统中,均会涉及到“文”的流转。一般的 OA 系统会涉及到两类流转方式:一类是符合政府、机关、学校等传统办文模式的“点对点流转”;另一类是符合企业、单位等业务流程需要的可定制的“流水线”式流转,主要用于协作办公。框架中设计了两类流转的通用模式,并且能通过配置文档灵活地与具体的“文”进行关联,从而实现流转功能。

“点对点流转”流转过程强调“按部就班”的特性。框架设计中利用 XML 数据格式来抽象流转过程中的每一节点,并作流程数据保存(“文”内容的易变部分),

以增加系统的通用性。每一个节点允许办“文”人员增加两类处理内容:一类是简单文本,另一类是独立文件。为了增加系统的保密性,在设计节点数据类型时设计了附属角色属性,用于控制办“文”过程中办“文”人员对其他人员的处理内容的“阅读”权。系统通过封装流程对象来处理相关的细节,一个“文”在流转过程中实现为不同的对象,都复合封装了流程对象(Flow),由该对象提供流转中需要的各种功能,并记录各种流转信息。

图2的简化操作序列说明了几个核心方法的交互过程。

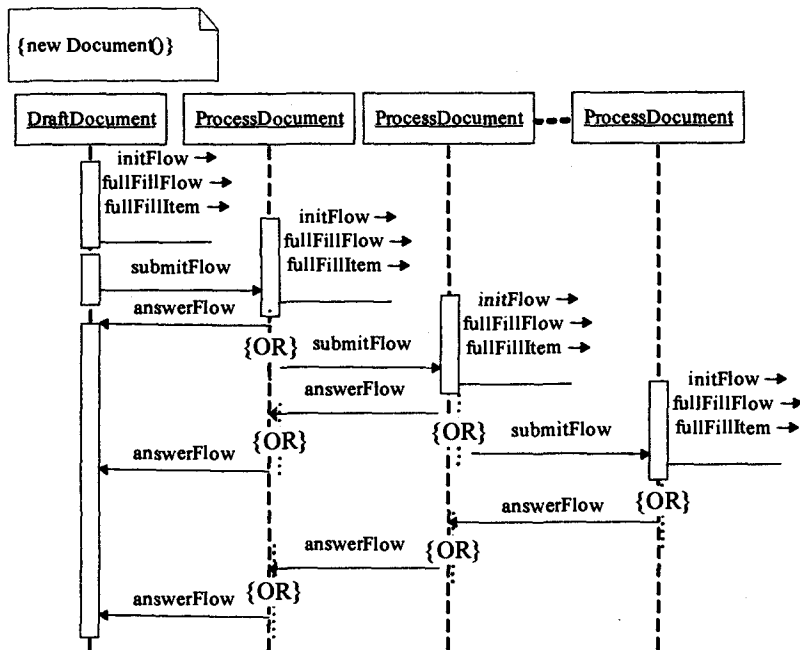


图2 流程核心方法交互示意图

“流水线”式流转过程强调流程的可配置性和控制性,框架中的主要处理内容为:在初始化流程时允许定制与该“文”绑定的流程内容或采用预定义默认流程设置,流程的每一个步骤可以是一个人、多个人或一个服务器方程序,甚至是服务器方的多个程序,并且在流转过程中,允许有权限的用户改变初始流程设置。框架设计利用 XML 数据形式来抽象此类流转,与“点对点流转”方式的抽象不同,“流水线”式流转的设计重点是流转步骤,每个步骤由若干节点组成,步骤对应为办“文”的阶段,节点对应为办文的人员(或程序)。步骤通过两个方法来描述:并行递交和串行递交。两个方法的执行由服务器方控制程序自动分析预定义的流程配置内容后自动调用,框架设计利用“反射”机制来实现该执行机制;节点用与“点对点流转”方式相似的方式记录了节点的数据,即处理内容。框架中设计了服务器方的自动程序,用于“文”的分发和整理。

框架中设计了一类独立的数据存储来保存流程配

置信息,利用消息触发方式来控制“文”的流转,所有与流程过程处理相关的代码自成一体,设计成一个完全独立的服务模块。

2) 监控设计。

在 OA 系统中,监控信息一般有两类:一类是提供给管理员的特殊信息,如监控系统的运行状态;另一类是“文”的处理状态信息,实现“文”的痕迹管理,跟踪“文”的状态。针对后者,框架中设计了一类独立的数据存储来保存此类监控信息,并设计独立的信息采集程序。信息的采集采用异步消息方式,由信息采集程序自动归类各种消息并合并,最终对于一个“文”,在该

类存储中仅有有一个记录与之对应。框架中设计了消息的标准格式,并设计了一个产生此类格式消息的 helper 类,在需要记录信息的功能操作中,把该类注册为一个事件监听器。

3) 保存操作。

此类操作是严格与用户视图绑定的,而且不同视图可能具有同名的操作,但实现的功能却不一样,因此,在定义“文”的格式视图时,单独定义了功能区(菜单与工具栏区域),此区域的内容来自于一个独立的配置文件,该文件详细定义了该视图可能具有的各种操作的详细信息,如操作的显示信息、执行的功能函数和执行顺序及此操作对应的权限设置等。在“文”进行显示时,无论是先期生成还是后期转换,由控制程序

分析该配置文件,并把相应功能进行关联等。

由于本框架的数据设计分成了相对不变部分和相对易变部分,因此视图操作的具体功能设计为由客户方委托给服务器方的 Transaction Service 来完成数据操作,视图操作代码仅记录了该操作的细节描述(XML 数据,一次操作称为一个 action),此类数据由客户方的 view Controller 维护,最终由 view Controller 把一次或几次用户操作动作(一次 Transaction)递交给服务器方的 Transaction Service,从而满足快速开发要求,同时也提供了一定的系统安全性和数据的完整性与一致性。

视图操作的具体的代码可以实现在“文”对象中,也可以是独立的 command 对象或脚本代码(B/S 模式)。同时视图格式定义了一些内部事件,如初始化事件(“文”显示时调用),并定义了对应的事件处理器,这些与功能操作不一样,具体的执行代码实现在“文”对象中,定义视图时只定义触发条件。

5 控制设计

框架的控制设计主要分为两类:一类是控制“文”的处理,一类是控制“文”的权限。后者是框架中安全访问设计的核心。

“文”的处理控制通过分析“文”对应的配置文档,正确形成一个“文”生命周期各个阶段对应的性质与状态,主要由初始化控制(产生一个新文档)、流转控制、显示控制、状态转换控制等组成。

“文”权限控制由两部分组成,物理操作属性控制(Access Control)和逻辑操作属性控制。物理操作属性控制决定一个用户对数据的物理操作,此类属性控制设计为系统级,框架中设计了一个校验服务,在用户涉及此类操作时提供操作许可校验,进一步加强数据的物理安全性;逻辑操作属性控制,采用了基于角色的访问控制机制(Role - Based Access Control, RBAC)^[6],框架中此类控制设计为与用户执行的功能操作相关,通过定义用户的属性文档(profile 文档)来定义用户允许的操作。用户视图的显示控制服务通过分析相应用户的 profile 文档提供用户准确的视图与操作(菜单项与工具栏项),服务器方的 Transaction Service 执行时也分析该文档,最终决定用户是否能执行该项功能。

6 使用和扩展

一个 OA 系统可以理解为若干“文”的一个集合,利用本框架开发一个 OA 系统时,主要任务是设计不同类别的“文”。一类“文”,一般仅需设计一个“文”数据源定义(定义本系统数据源时,应设计对应的 form 或 page)、一个流程与视图转换定义、若干视图格式定义(包含视图功能操作定义)、一个物理权限定义。同时配置一个全局定义文档,该文档定义了系统所有“文”的基本信息,主要是各个“文”的基本信息(名称信息,各个配置文档的基本信息,如路径信息等),同时在该配置文档中定义了每类“文”的角色配置,此处定义的角色信息主要用于配置用户 profile 文档和进行运行时权限校验。

对于视图操作,框架中提供的一个“文”实例的操作能满足一般应用,如需开发扩展功能,可以遵循以下描述:框架设计中对于视图操作的处理,根据其是否涉及服务器端数据操作采用两类不同的处理方式(Transaction Action 和 Single Action),对于前者开发时

需设计以下要素,一个产生操作描述的 Command 对象或脚本代码、与操作描述信息(方法、参数、顺序)对应的服务器方 Service(由 Transaction Service 调用)、在视图功能配置定义中声明该操作信息;对于后者,一般涉及到客户端的数据整理和视图的运行控制,仅需设计一个产生该功能的 Command 对象或脚本代码及实现操作所需的一些附属信息,同时在视图功能配置定义中声明该操作信息。

如需在服务器方增加更多的自动动能,则需设计实现该功能的 Agent 或 Service,同时配置一个与“文”绑定的默认服务器端流程处理文档(一种流程的特殊引用)。

7 结束语

文中的创新点是通过分析 OA 系统的内容及其处理特征,结合具体的项目实施经验和成熟的软件设计方法,设计和实现了一种基于配置管理的、符合快速开发要求的 OA 系统框架。

利用该框架具体部署一个 OA 系统时,能获得项目初期的快速开发能力、中期的灵活、快速部署能力、后期的简便维护能力,同时获得较高的系统整体安全性和集成能力,适用于高校、政府机关和中小企业的 OA 系统开发应用。

参考文献:

- [1] Husted T N. Struts in Action: Building web applications with the leading Java framework[M]. America: Manning Publications, 2002.
- [2] Sauter P. Extending the MVC Design Pattern towards a Task - Oriented Development Approach for Pervasive Computing Applications[J]. Computer Science, 2004, 2981: 309 - 312.
- [3] Chung S. A Web Service Oriented Integration Approach for Enterprise and Business - to - Business Applications[J]. Computer Science, 2004, 3306: 510 - 515.
- [4] 郭祖华. 基于 Web Service 的松散耦合分布式电子政务系统[J]. 微计算机信息, 2006, 12: 195 - 197.
- [5] Noga M. Optimizing Content Management System Pipelines Separation and Merging of Concerns[J]. Computer Science, 2002, 2487: 252 - 267.
- [6] 吕宜洪. 政府机关公文流转系统访问控制基本框架及改进的 RBAC 模型研究[J]. 计算机应用研究, 2003(1): 31 - 34.

欢迎订阅《计算机技术与发展》邮发代号: 52 - 127