

基于 J2EE 架构的办公自动化系统的设计研究

叶宇风 胡贞华

(韶关学院网络中心 广东韶关 512005)

摘要 通过对 J2EE 架构的分析,提出了基于 J2EE 体系的办公自动化总体架构和子系统的分块,然后分别阐述了各子系统的功能及设计方法。

关键词 J2EE 架构,办公自动化,设计

Study of Designing of OA System Based on J2EE Frame

YE Yu-Feng HU Zhen-Hua

(Network Centre of Shaoguan University, Guangdong Shaoguan 512005)

Abstract Based on the analysis of J2EE Frame, the article proposes the general frame of OA system based on J2EE, and elaborates each subsystem's function and design method.

Keywords J2EE frame, OA, Design

在企业信息建设的进程中,随着网络硬件基础设施建设规模的扩大和日趋完善,网上的各种应用和网上资源的有效整合将摆在越来越重要的位子。当前,作为企业信息化建设重要组成部分的办公自动化系统,多数还是采用早期的 C/S 传统模式设计,系统的应用虽然给工作带来便利,取得了成效,但也暴露出了一些普遍存在的问题:系统存在较大的安全隐患;要实现与其他应用系统的有效整合和更大范围的信息共享,难度较大;系统升级和维护的成本高;不能很好地适应应用需求的变化。为此,本文从研究 J2EE 体系结构的特点出发,介绍了在网络环境下基于 J2EE 体系架构,该架构便于资源整合,能方便、直观地适应各种业务处理流程变化的办公自动化系统的设计方法。

1 J2EE 体系框架及其特点

J2EE(Java2 Platform Enterprise Edition)体系结构(图 1)是 SUN 公司于 1999 年底主持推出的一项企业计算平台规范,是一种利用 Java 平台来简化诸多与多级企业解决方案的开发、部署和管理复杂问题的体系结构。J2EE 定义了包括平台角色、组件模型、标准服务等一系列规范。它具有“编写一次、随处运行”的特性、方便存取数据库的 JDBC^[1] API, CORBA, JNDI, RMI 技术以及能够在 Internet 应用中保护数据的安全模式等功能,为用户和开发人员提供一个具有高度的可移植性、兼容性和安全性^[2]的平台。J2EE 系统一般是由客户层、Web 层、业务层、数据库层和企业信息系统层(EIS)组成,其中 Web 层和业务层又可统称为中间层,如图 1 所示。

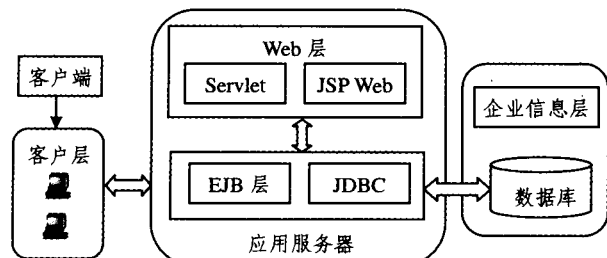


图 1 J2EE 的多层架构

1.1 客户层

客户层由浏览器或其他一些基于网络的系统(如手机和 PDA 等)组成。用户可通过客户层从服务器上访问 Web 层的静态 HTML 页面或由 JSP 或 Servlet 生成的动态页面和获取所需的信息。客户层可以是传统的客户端程序,也可以是运行在网络设备之上的 Applet 小程序,还可以是不通过 Web 层而直接访问业务层中的 EJB(Enterprise JavaBeans)^[3]。

1.2 Web 层

Web 层可以由 JSP 页面、Applets 或 Servlets 组成,通过处理客户层上的用户输入请求,并把请求消息发送给运行在业务层上的 EJB 进行处理。JSP(Java Server Pages)是一种动态 Web 应用技术标准,是由传统 HTML Web 页的文件(*.html, *.htm)中加入 Java 程序片(Scriptlet)和 JSP 标记构成的,可以在服务器端根据客户请求产生动态页面。Servlets 是服务器端的 Java 小程序,通过 Servlets API 响应客户端的请求并进行处理,可以产生动态的 HTML 页面。Web 层和业务层通常被封装在一个应用服务器中。

1.3 业务层

业务层又被称为 EJB(Enterprise JavaBeans)层。EJB 定义了一个面向对象的可重用服务器端分布组件标准,用来封装业务逻辑。EJB 是 J2EE 体系结构的核心,在功能上分为会话 Bean(Session Bean)和实体 Bean(Entity Bean)。Session Bean 执行事务逻辑、算法、规则和工作流程,是具有处理事务过程逻辑的可重用组件。Entity Bean 将底层数据以对象的形式映射到内存中,供其他组件使用。这种将事务逻辑与底层数据分离的方法,使得应用能够适应不同的操作系统,增强了应用系统的可移植性和可扩展性。

1.4 企业信息系统层

企业信息系统层主要是处理信息系统软件,如:基础建设系统、数据库系统和其他的应用系统等。J2EE 的多层次结构的特点,能够为开发应用系统提供分布式及开放式的支持,在实施应用程序部署时可以实现组件重用。

J2EE 架构平台能够提供以下的服务功能:

1)安全管理。通过配置 Web 组件和 Enterprise Bean,使得被授权的用户才能访问系统资源。每一客户属于一个特定的角色,而每个角色只允许激活特定组件。因此,通过在 Enterprise Bean 的部署描述中声明角色和可被激活的方法可以大大加强系统的安全性,而开发者不需要编写加强安全性的规则。

2)事务管理。J2EE 可以指定组成一个事务中所有方法间的关系,将一个事务中的所有方法当成一个单一的单元进行处理。当客户端激活一个 Enterprise Bean 中的方法,则由事务管理模型加以管理。这些方法要么全部完成,要么全部取消。

3)远程方法调用^[3]。J2EE 通过 RMI 模型管理客户端和 Enterprise Bean 间的底层交互,当一个 Enterprise Bean 创建后,客户端可以调用它的方法,就像它和客户端位于同一虚拟机上一样。

4)数据库连接池。J2EE 通过数据库连接池模型管理来获取数据库的连接。获取数据库连接是一项耗时的工作,并且连接数量非常有限。Enterprise Bean 可以从池中迅速获取连接和有效释放连接供其他 Bean 使用,通过管理连接池会大大地提高系统的运行效率。

5)完全支持模块化开发。J2EE 能以 Java Archive(JAR)文件的形式交付。组件的可重用性使得将应用程序的设计、开发、封装、部署等过程划分给不同的角色。这样,不同人或机构就能独立完成整个过程的不同部分,提高开发效率。

应用 J2EE 开发办公自动化系统的优势主要有以下几点:

(1)基于 J2EE 可构造典型的多层应用,不同的业务角色和人员通过浏览器从不同的地方访问系统,在业务层中处理复杂的业务逻辑,在数据库层储存用户、规则等相关信息。多层结构机制与有效数据库安全技术保证系统高可靠性。

(2)开发系统能集成原有信息系统,并保证不同系统间的互操作性,有利于系统的模块化设计,减少应用层中与其他模块的耦合,符合当今软件工程开发规范。

(3)开发的程序可以运行在多个操作平台与应用平台之上,符合开放性原则。

(4)可以充分利用 J2EE 各项技术开放程序,可以提高开发效率,降低维护成本。

(5)在技术上具有先进性,能够为新增加的应用和需求变化提供系统支持^[3]。

2 基于 J2EE 架构的办公自动化系统的设计

办公自动化系统实质上是基于不同部门、角色、用户基础上的多层次、实时性和安全性要求高的一种应用系统。这种应用系统的关键是既能够适应机构、部门、人员和业务处理流程的经常性的变化,又能够为新增的应用服务提供良好的接口,即能够完全适应应用变化的需求。J2EE 多层架构,能够很方便地将办公自动化的实际应用需求模块化,并确定模块之间的关系,通过创建各类 EJB 应用组件来构建整个应用系统。

如图 2 所示,办公自动化系统由用户管理、公文管理、信息管理、模块管理、安全管理等功能子系统构成。

2.1 用户管理子系统

1)机构配置管理。机构配置管理主要是配置单位各职能

部门的组织机构,通过定义各职能部门的属性及其相互间的关系,形成树状组织机构图。在机构配置中,通常包含的项目有:部门名称、简称、编号、排序号、所属部门、部门的属性及部门的一些描述性内容,如部门简介、职能、联系电话等。机构配置管理将实际组织机构映射到系统中,为系统按实际组织结构进行授权打下基础;通过将组织部门进行统一分类、编号,为系统中生成档号、文号等提供基准。

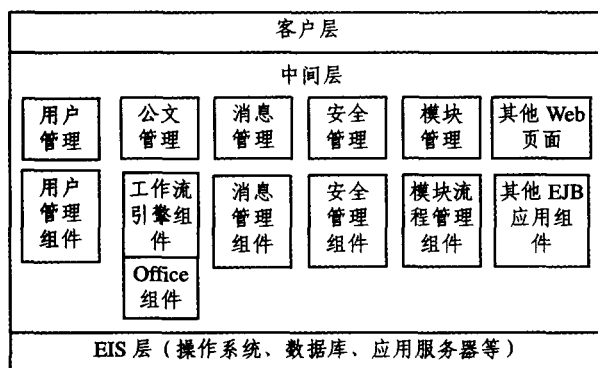


图 2 办公自动化系统结构

2)人员配置管理。人员配置主要定义人员属性和人员基本信息。人员配置主要包含:人员名称(与 ID 一致)、所属部门、角色和个人的一些基本信息。人员角色的定义可根据实际情况定义,可通过将多人设置成同一角色或一个人设置多个不同角色的方法,实现一岗多人或一人多岗的描述。角色的定义与使用权限管理相关联,授权通常要到角色。

2.2 公文管理子系统

公文管理关键是工作流引擎,工作流引擎子系统用来描述各类繁杂的办公业务流程,通过定义流程节点属性、操作属性等元素将业务流程模型化。从而方便地对办公业务流程进行描述、更改,达到快速搭建不同办公业务系统。工作流引擎主要包括两个配置项,即流程节点配置和路由配置。

1)流程节点配置。用于定义业务流经过的各个环节,通过节点属性描述该业务环节的具体功能及权限等内容。节点属性通常包括节点名称、编号、操作角色、编辑操作权限等。

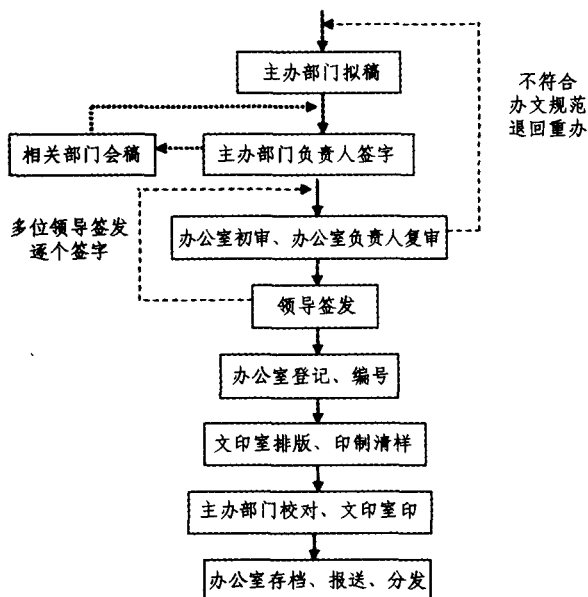


图 3 发文流程图

2)路由配置。用于控制流程的流向,不同的路由指向不同的操作节点,通过操作属性选择下一步的流向,同一节点可以单一操作或多个操作,操作属性通常包括操作名称、编号、源节点、目标节点、涉及范围对象、提醒方式(文件到来时的提醒方式,选项包括邮件、网络寻呼、短信息等)。

流程配置的步骤,首先定义点(流程节点),其次通过单向线(路由)将点对点进行连接,然后再定义分支与循环关系最终形成流程图。一般来说,流程配置比较复杂,容易出错,容易形成断点。可通过设计可视化流程配置工具(图3以发文流程图为例),直观反映流程图的整体情况,并与通过检查工具确保流程配置的正确性。

在公文管理中,除了复杂的业务逻辑之外,还必须考虑到用户的工作习惯及公文传输过程的留痕等问题,需要提供与Word、Excel、WPS等流行常用的Office软件无缝连接。在公文流转过程中,同一公文提交多人同时办理或依次办理时,能够自动地对每次操作进行控制,并将多人处理结果进行有机整合,通过留痕管理对文件的修改过程进行全方位跟踪,对修改人、修改时间作出明确的记录。

2.3 消息管理子系统

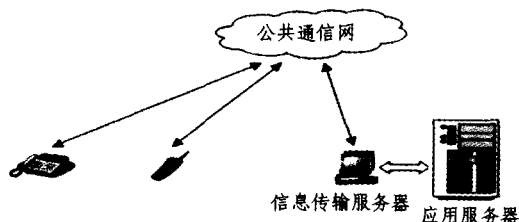


图4 消息传输图

在传统办公自动化系统中,由于受通信条件的限制,用户的待办事宜唯有通过被动的查询方式获得业务的办理情况。消息管理子系统主要是利用当前能够提供公共通信服务,如:电信、移动等提供与手机、电话通信的功能,提供手机短信提示、电话提醒等功能。使待办公文能够主动及时地通知办理者,从而提高公文办理效率。消息管理子系统除了对登录用户实时提醒外,还能够通过手机、固定电话等通信工具通过公

共通信网与本地的信息传输服务器、应用服务器系统连接,查询到相关信息的功能。如图4所示,信息传输服务器主要是按照传输协议处理公共通信网与内部应用系统之间的通信和数据转换。

2.4 模块管理子系统

模块管理子系统主要用于组织模块的层次结构。通过定义模块的分组、分层结构,使各模块按照配置好的层次加以展现。模块配置管理组件化可以增强系统的灵活性和可扩展性,可以快速将二次开发完成的模块或组件发布到系统中。模块管理主要包括如下配置项:模块所属目录、模块所在服务器、模块位置等。

2.5 安全管理子系统

安全管理子系统具备统一的身份认证、统一的门户管理、多重安全保护等功能。通过对数据链路层、网络层的信道加密和数字证书应用管理系统(或与第三方PKI/CA体系)进行身份认证、访问和系统的监控、日志分析,保证办公自动化应用系统能够方便地管理和访问不同的硬件平台、操作系统、网络协议和数据,使系统的安全风险降到最低限度,从而实现“单点登录,Web应用漫游”。

结束语 上述办公自动化系统已在某地方高校开发成功,投入使用。该系统具有完备的接口和良好的耦合性,与按照教育部信息化建设标准建立的学校基础数据库连接,实现了与以前开发的应用系统有效整合。实际应用表明,应用J2EE多层次架构开发的办公自动化系统具有很好的开放性、实用性、安全性、可扩展性和可维护性,应用效果良好。

参考文献

- [美]Horstmann C S, Cornell C. Java2 核心技术(第2卷):高级性能[M]. 北京:机械工业出版社,2003
- [美]Horstmann C S, Cornell C. Java2 核心技术(第1卷):基础知识[M]. 北京:机械工业出版社,2003
- [美]KEOGH J. J2EE 参考大全[M]. 宁建平,等译. 北京:电子工业出版社,2003. 9~25
- [美]Couch J. J2EE 宝典[M]. 马琳,等译. 北京:电子工业出版社
- Oppenheimer D, Ganapathi A, Patterson D A. Why do Internet services fail, and what can be done about it?. In: 4th Usenix Symposium on Internet Technologies and Systems(USITS '03), 2003
- Ganapathi A. Failure Analysis of Internet Services. Computer Science Division, University of California, Berkeley
- Sun Microsystems. <http://java.sun.com/j2ee/>
- Microsoft; <http://www.microsoft.com/net/>
- Omg; <http://www.omg.org/technology/documents/formal/components.htm>
- Szyperski C. Component Software: Beyond Object-Oriented Programming. Addison-Wesley, 2002
- JBoss; www.jboss.org/
- Diaconescu A, Mos A, Murphy J. Automatic Performance Management in Component Based Software Systems. Performance Engineering Laboratory, Dublin City University
- Patterson D. Recovery Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies. University of California at Berkeley
- Candea G, Kawamoto S, Fujiki Y, et al. Microreboot: A Technique for Cheap Recovery. In: Proc. of the 6th Symposium on Operating Systems Design and Implementation(OSDI), Dec 2004
- Candea G, Fox A. Recursive Restartability-Turning the Reboot Sledgehammer into a Scalpel. In: Proc. of the 8th Workshop on Hot Topics in Operating Systems(HotOS-VIII), May 2001
- 万群丽,杨群,李正,许满武. 一种基于 Agent 适用于 Web 应用的软件抗衰方法,计算机应用研究,2004,21(8):18~21,26
- Sullivan M, Chillarege R. Software defects and their impact on system availability: a study of failures in operating systems. In: Proc. 21st Intl. Symposium on Fault-Tolerant Computing, Montreal, Canada, 1991
- Gray J. Why do computers stop and what can be done about it?. In: Proc. 5th Symp. on Reliability in Distributed Software and Database Systems, Los Angeles, CA, 1986
- Chou T C. Beyond fault tolerance. IEEE Computer, 1997, 30(4):31~36

(上接第277页)