

差旅报销系统及其 workflow 引擎的设计与实现

过 怡

(苏州职业大学 江苏 苏州 215011)

摘 要 针对企业传统的差旅报销系统,根据系统整合的建设目标,提出了差旅审批和费用报销系统的业务流程分析,Workflow 引擎设计和系统总体设计、功能设计,并以 Workflow 技术为基础,以基于 Web 应用为背景,给出了差旅审批和报销系统的解决方案。该方案有较强的实用性和通用性,提高了企业科学决策的能力,为类似系统的设计和开发提供了思路和方法。

关键词 差旅审批 报销 业务流程 Workflow 解决方案

DESIGN AND IMPLEMENTATION OF TRAVEL AUTHORIZATION AND EXPENSE CLAIMANT SYSTEM WITH WORKFLOW ENGINE

Guo Yi

(Suzhou Vocational University, Suzhou 215011, Jiangsu, China)

Abstract The business process of enterprise-level travel authorization and expense claimant system is described, and the workflow engine and the functional modules of the system are designed. A Web-based solution to travel authorization and expense claiming with workflow technology is provided. The decision capability of enterprises is improved, and the proposed scheme has provided a way to improve similar business process.

Keywords Travel authorization Expense claiming Business process Workflow Solution

0 引 言

有效整合企业信息资源以适应灵活多变的需求是企业发展和持续提高竞争力的重要方法,但很多企业的信息化是不完善的。其中在信息化的一个很重要的领域——出差审批和报销业务中,延续了数十年的传统模式依然罕有变化。

传统的出差审批和报销模式采用手工的方式来实现。员工每次报销,填写纸面的单据并拿着单据找到各级领导进行审批,最终将单据拿到财务进行审核。我们可以看到传统的模式存在下述诸多问题:

- (1) 单据填写效率低下。
- (2) 业务审批效率低下。
- (3) 财务审核效率低下。
- (4) 无法进行有效的资金控制。
- (5) 无法提取有价值的信息。
- (6) 内部控制不力。

针对以上问题,研究出差审批和财务报销的业务流程,实现企业对出差审批和财务报销的全过程控制,提供系统解决方案,具有重要的现实意义。

1 系统目标

本文以出差审批和报销的通用流程为基础,以 IT 基础架构标准库(ITIL)环境下系统的建设为指导,提出本系统的目标:

- (1) 提供完善的出差审批和报销流程。

- (2) 提供完善的现金预付\现金支付体系。
- (3) 提供完善的安全控制机制。
- (4) 提供灵活的报表分析功能。
- (5) 考虑与企业 ERP(mySAP)系统的数据集成,实现信息共享
- (6) 提供用户友好的界面。

2 系统分析

2.1 业务流程分析

出差审批的基本原则是依据公司组织架构逐级审批,对应于不同的目的地决定是否需总部副总裁的审批。企业的审批以成本中心为基本单位,每个成本中心都有其不同层级的领导人和相关财务行政人员。企业对于出差审批的一般流程如表 1 所示。

表 1 差旅审批的一般流程

审批层级	审批规则
计费成本中心经理	必需
申请人成本中心经理	仅当申请人成本中心 < > 计费成本中心时需要
申请人成本中心秘书(仅需通知)	必需
培训主管(仅需通知)	必需
财务主管	仅当有差旅预支款时需要

收稿日期:2007-03-09。过怡,硕士,主研领域:计算机应用技术。

申请人成本中心会计(仅需通知)	仅当有差旅预支款时需要
计费成本中心总监	必需
计费成本中心副总裁	仅当目的地在亚洲之外时需要

费用报销审批的基本原则是依据企业组织架构逐级审批,对应于不同的报销金额决定是否要上一级管理人员审批,并引入相关部门秘书和财务审核。费用报销的一般流程如表 2 所示。

表 2 费用报销的一般流程

审批层级	审批规则
申请人成本中心秘书	必需
计费成本中心主管	必需
计费成本中心副经理	必需
计费成本中心经理	仅当数额 > \$ 1000 时需要
计费成本中心总监	仅当数额 > \$ 5000 时需要
计费成本中心副总裁	仅当数额 > \$ 50000 时需要
申请人成本中心会计	必需

如果成本中心中审批层级没有相对应的审批人,该流程需要自动向上升高审批层级。从以上流程可以看出,审批流程涉及的人员多,人员层次复杂,业务量巨大,系统解决方案的核心即在于此。

2.2 工作流引擎

为了解决复杂而又灵活的审批流程,我们依据 WfMC(国际标准化 Workflow 组织)所提供的 Workflow 参考模型^[1-3],以组件形式设计和封装了系统核心的 Workflow 引擎,使其与数据库和用户界面分开^[4]。Workflow 引擎又由接口和内部核心处理两部分组成,在内部核心处理中,采用了缓存和多线程等技术来提高系统性能。

该 Workflow 引擎基于 .Net 平台开发,开发环境是 Windows 和 MS SqlServer,采用 C#开发语言,利用数据持久化组件进行数据库增、删、改、查询等 CRUD 操作,使开发人员能够专注于业务流程的实现上。

(1) 接口

包括流程管理接口和事件触发接口,流程与具体应用进行连接的接口。流程管理接口如图 1 所示,包括流程的定义和实例两部分的接口。WorkflowCats 类, Tasks 类和 Actions 类是 Workflow 引擎的核心,WorkflowCats 类集中了不同类型的 Workflow,负责内部节点 Node 的定义和 Workflow 的创建,免去了客户端创建的责任,实现了责任的分割。管理员每定义一个新的应用流程,产生一个 Workflow。Tasks 类产生一个 Task(Workflow 实例),其中每进行一步处理,产生一个 Proc。

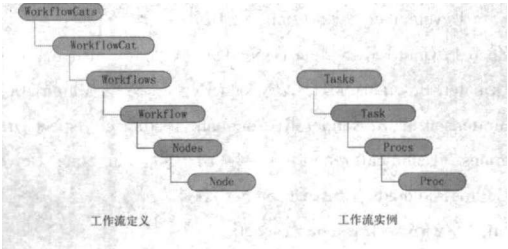


图 1 流程管理接口

每个 Workflow 实例中各节点用户在处理过程的 Proc 有其

相应的事件触发接口。如图 2 所示,包括对流程的启动、处理、退回和分派等动作处理的接口。通过这些接口,系统可以计算下一步的处理走向,整个流程的状态和触发相关的事件。

流程与具体应用进行连接的接口如图 3 所示,生成新的 Workflow 实例时,调用本接口。当有新的表单提交时,系统通过 TaskName 和 Param、Poster、Receptor 等参数,产生一个新的 Task,即 Workflow 实例及其相应节点,等待下一节点用户进行处理。

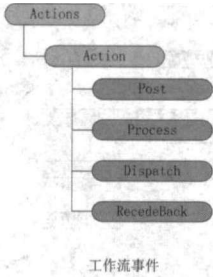


图 2 事件触发接口

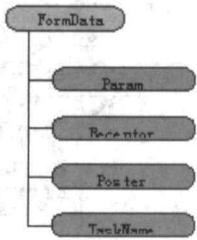


图 3 Workflow 与应用连接的接口

(2) 内部核心处理

内部核心处理有流程处理和事件处理。流程处理包括并行、串行、挂起、退回、重起等一系列的流程流转处理。当前节点用户处理(Proc)提交时,根据当前流程的定义和实际处理所提交的参数,计算下一步的处理走向、整个流程的状态和相关 Email 的发送等动作。流程的每一步,用户可定义参数包括:

- (i) 处理节点名称。
- (ii) 类别:并签,串签和跳签。
- (iii) 条件参数:可根据用户填写表单时所填的内容来决定流程的走向。
- (iv) 处理者:可以是具体的人,或各种角色。
- (v) 处理者所能选择的事件(动作):如退回、保存等。

事件处理分人工触发事件和自动触发事件的处理。事件分三种类型:提交到后续节点,退回到先前节点和保存在当前节点。

综上所述,该引擎通过事件驱动生成 Workflow 实例,通过 Workflow 流程管理接口和事件触发接口进行核心处理,能比较灵活地解决审批流程的需求。

3 系统总体设计

差旅报销系统是 Workflow 引擎的一个应用实例。根据系统的需求和用户分散的特点,我们采用基于 Web 的方式平台进行整体设计,技术平台采用 .Net, C#和 MS SqlServer,通过调用 .Net 组件方式集成 Workflow 引擎,在 Workflow 引擎的基础上将出差审批和报销系统分为五个子系统:出差审批子系统、费用报销子系统、现金预付子系统、现金支付子系统、报表子系统。

差旅报销系统系统主要包括出差申请以及审批。费用报销子系统包括出差费用和一般费用的申请以及审批。现金预付/现金支付子系统是相对静态的系统,依据出差项目或报销项目预付和支付现金。报表子系统为管理人员提供业务报表。

3.1 系统结构

整个系统可分为 4 个层次,如图 4 所示。

- 1) 界面层:它是系统同用户进行交互,获取或输出信息的接口。它通过提供直觉化的界面,用户可以方便地提出要求,得

到所需的信息。

2) 功能层:它由执行出差审批和报销的各个子系统构成。各子系统又包括若干业务,这些业务可构成业务层,业务层之间既有顺序的、又有并列的。

3) 中间件层:这里指 Workflow 引擎,控制审批流程。

4) 支持层:它是指系统所用到的数据库管理系统、操作系统和网络通信协议(如 TCP/IP)等,是保证整个系统正常运行的基础。

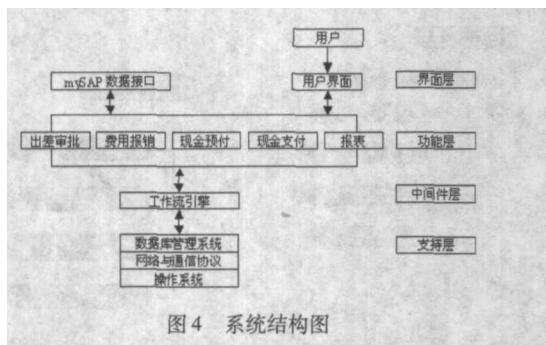


图4 系统结构图

3.2 运行模式

根据出差审批和费用报销企业集团中的应用要求,考虑到企业内部信息管理以局域网为主体,系统需要实现多用户、不同区域间的实时合作,信息交互。因此系统可以充分利用基于 Web 的浏览器/服务器模式的优点,保证内部信息的高度安全及信息传送和处理的高效。当用户处于局域网外时,可以随时随地通过 VPN(Virtual Private Network)登录局域网安全使用系统。因此,运用 VPN/Intranet 相结合的技术方案,实现系统的最佳性能。如图 5 所示。

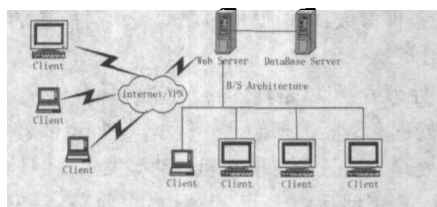


图5 系统的运行模式

4 结论

本文针对传统的出差审批和报销模式,分析了出差审批和报销的业务流程,建立了系统结构,讨论了 Workflow 引擎的接口和内部核心处理,设计了基于 Workflow 的解决方案。该方案可以实现灵活的流程控制,具有较强的实用性和通用性,能有效地提高企业的信息化水平和科学决策能力。为类似的系统要求提供了一种思路。

参考文献

- [1] The Workflow Management Coalition. Workflow Standard-Interoperability Abstract Specification[M]. Avenue Marcel Thiry 204 1200 Brussels Belgium, 1996:23-61.
- [2] The Workflow Management Coalition. Workflow Process Definition Interface-XML Process Definition Language[M]. 2436 N. Federal Highway #374 Lighthouse Point, FL 33064 USA, 2001.
- [3] The Workflow Management Coalition. WIDE Workflow Development Methodology. University of Twente, 1999.
- [4] 李海波,战德臣,徐晓飞. 基于工作流引擎的构件组装体系结构[J]. 软件学报, 2006, 17.

- [5] 范玉顺. 工作流管理技术基础[M]. 北京:清华大学出版社, 2001: 63-66.
- [6] 向俊莲,杨杰,梅宏. 基于软件体系结构的构件组装工具 ABC-Tool[J]. 计算机研究与发展, 2004, 41(6):956-964.

(上接第 148 页)

主要是测试方面中方法以及内部声明,不过不完全自动,其中还须人工的操作。Y. Yamazaki^[7]提出了基于无编织的面向方面单元测试框架,不过只是针对简单的面向方面程序以及简单 AspectJ 关键词进行测试。

Zhao^[8-10]提出了基于数据流图的单元测试面向方面程序的思想以及提出构造面向方面系统依赖图。

Mort^[11]讨论了结合两种传统的测试技术:结构化测试方法和仿真测试方法,对面向方面程序进行测试。方面被分成是否修改系统状态两种,这种技术主要是去识别方面中的方法或通知引入的错误。

6 结论

本文基于 AOP 的特性和方面之间的交互错误模型的研究,提出了 Aspat 框架去识别方面交互模型中可能出现的错误,并给出一定的面向方面编程经验。未来将对 Aspat 进行扩展,使其能识别更多的错误。

参考文献

- [1] Gregor Kiczales, John Lamping, Anurag Mendhekar, et al. Aspect-Oriented Programming. Proc. the European Conference on Object-Oriented Programming, Finland, Springer-Verlag, LNCS 1241, June 1997.
- [2] Ramnivas Laddad. AspectJ in action Manning publishing. 2003.
- [3] Dahm M, van Zyl J, Byte Code Engineering Library. April 2003. <http://jakarta.apache.org/bcel/>.
- [4] Junit: <http://www.junit.org>.
- [5] <http://www.jmock.org>.
- [6] Tao Xie, JianJun Zhao. A Framework and Tool supports for Generating Test Inputs of AspectJ Programs. AOSD06, March 20-24, 2006.
- [7] Yamazaki Y, Sakurai K, Matsuura S, et al. A Unit Testing Framework for Aspects Without Weaving. Proc. 1st Workshop on Testing Aspect-Oriented Programs, March 2005.
- [8] Zhao J. Data-Flow-Based Unit Testing of Aspect Oriented Programs. Proceedings of the 27th Annual IEEE International Computer Software and Applications Conference, Dallas, Texas, USA, November 3-6, 2003:188-197.
- [9] Zhao J. Tool support for unit testing of aspect-oriented software. in Proceedings OOPSLA' 2002 Workshop on Tools for Aspect-Oriented Software Development, November 2002.
- [10] Zhao J, Rinard M. System Dependence Graph Construction for Aspect-Oriented Programs. MIT LCS Tech Report 891, March 2003.
- [11] Mortensen M, Alexander R. Adequate Testing of Aspect-Oriented Programs. Technical report CS 04-110, Colorado State University, Fort Collins, Colorado, USA, December 2004.
- [12] <http://www.eclipse.org/aspectj>.
- [13] Xerox PARC Aspect-Oriented Programming with AspectJ 1.0.1 AspectJ.org.