

基于 Web 的分布式 workflow 系统的研究

姜晨临, 蔡鸿明

(上海交通大学 软件学院, 上海 200030)

摘要: 在 workflow 技术研究领域中, 基于 Web 的分布式 workflow 系统逐步成为研究热点。介绍了一种基于 Web 服务技术的分布式 workflow 管理系统的框架结构, 并从数据层、业务层、应用层三个方面详细介绍实现此 workflow 执行框架所使用的各种技术, 研究了在这个框架结构中的核心部件——嵌入式 workflow 引擎的系统设计模型, 在该模型中探讨了数据模型和控制模型的设计。

关键词: Web 服务技术; workflow 管理系统; 分布式 workflow; workflow 建模; workflow 引擎

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2008)06-0043-04

Study of Distributed Workflow System Based on Web

JIANG Chen-lin, CAI Hong-ming

(School of Software, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: In the workflow technology study field, the study of distributed workflow system based on Web is a hot point to be studied. Introduces a framework of distributed workflow management system based on Web service, discusses all kinds of technique of realizing this workflow in detailed from data layer, business layer and application layer, and points out the core of this framework, that is the design model of inserted workflow engine system. In this model, it discusses the design of data model and control model in details.

Key words: Web service technology; workflow management system; distributed workflow; workflow model; workflow engine

0 引言

在当前 workflow 技术的研究领域中, 分布式 workflow 系统已逐步成为研究的热点。由于 Web Service 是一种构建应用程序的普遍模型, 它可以在任何支持网络通信的操作系统中实施运行, 只要用户能生成和使用 Web Service 接口所规定的 Message, 便可以在任何平台上通过程序化语言来执行 Web Service^[1]。因此, 文中提出了一种基于 Web Service 多层结构方式的分布式 workflow 系统实现框架, 介绍实现此 workflow 执行框架所使用的各种技术, 研究了在这个框架结构中的核心部件——嵌入式 workflow 引擎的系统设计模型。

1 基于 Web 的分布式 workflow 执行系统体系结构

在文中的 workflow 管理系统中, 底层通信基础结构是基于 TCP/IP 的 XML 格式的 Web 服务, 整个系统的体系结构如图 1 所示。

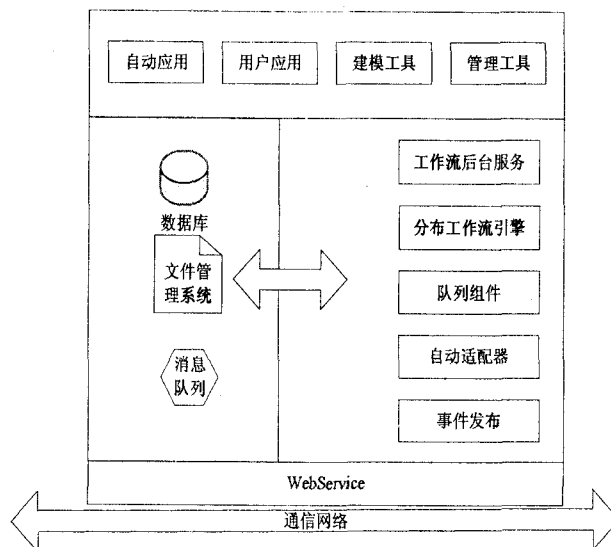


图 1 基于 Web 的分布式 workflow 系统实现框架 (1) 数据层。

数据层是 workflow 系统的数据基础, 由数据库、文件管理系统、消息队列三个服务器组成。其中数据库保存 workflow 系统中的模型数据、实例数据和日志记录, 文件管理系统辅助数据库保存文件类型数据, 消息队列则用来存储和管理 workflow 系统中的业务事件信息。在 workflow 系统中引入业务事件的概念使得在过程实例内部或不

收稿日期: 2007-09-25

作者简介: 姜晨临(1974-), 男, 硕士研究生, 研究方向为物流与企业信息系统。

同的过程实例间,甚至不同 workflow 系统间可以通过业务事件方式进行交互。

(2) 业务层。

业务层是 workflow 系统功能实现的核心部分,主要包括分布式 workflow 引擎、workflow 后台服务、自动应用适配器、队列服务组件和事件发布五个基本模块,采用 Web Service 的方式进行封装。下面主要介绍 workflow 引擎和 workflow 后台服务两部分的内容。

a. 分布式 workflow 引擎是该层的核心组成模块,它通过将 workflow 相关业务逻辑封装成为 Web Service 服务器的方式,为逻辑层其他模块和应用层提供访问和操纵数据层的标准接口。

b. workflow 后台服务是 workflow 系统在执行服务接口基础上提供的一套功能性辅助服务,主要负责实现时间管理、条件管理、自动应用管理、组织资源分配等功能。

(3) 应用层。

应用层由建模工具、客户应用、自动应用和管理工具组成。其中客户应用和自动应用还包含在系统实施过程中针对具体业务内容所定制开发的应用,这些应用都将运行在 workflow 系统的集成环境中。应用层对 workflow 系统数据的操纵,都是通过 workflow 引擎提供的接口实现的。

在 Web Service 技术提供了强大的应用分布于底层通讯的基础上,三层机构很好地实现了 workflow 系统中应用、逻辑与数据的分离。应用与逻辑的分离规范了应用对数据的操作,增强了系统的可重用性,使得应用开发的重点放在与用户的交互之上;逻辑与数据的分离优化了数据的共享机制,增强了数据的安全性。

整个系统都是采用面向对象方法设计、实现的,使用 Java 语言实现具体代码。系统的支撑系统包括数据库服务器和 Web 服务器,数据库服务器采用 MySQL, Web 服务器采用开源的 Tomcats。从实现角度来看,系统采用了三层结构,用户通过 Web 浏览器或其他客户端访问 Web 服务器,然后通过 Web 服务器中的 JSP 来访问数据库并实现业务功能。

2 工作流引擎的设计

在框架中介绍的工作流引擎是基于关系的。所谓基于关系的工作流引擎指的是 workflow 引擎中的数据模型(即机构模型和数据模型)全部通过关系结构来表达;控制 workflow 引擎运作的各种程序逻辑(即控制模型)也是通过常规关系数据库管理系统中所提供的存储过程、包以及触发器等机制来实现;同时,事务的并发控制也通过数据库系统所提供的机制来实现^[2]。

2.1 数据模型设计

基于关系结构的轻量级 workflow 引擎的数据模型包括机构模型和信息模型两部分。机构模型描述的是企业或者部门的组织机构关系,信息模型则定义 workflow 引擎中所用到的各种控制数据。通过数据模型,可以方便地描述关键业务的业务规则、活动的依赖关系以及任务的指派等特征。它们都通过统一的关系结构来定义。图 2 给出了基于关系结构的轻量级 workflow 引擎的数据模型的 E-R 图(限于篇幅只给出核心表结构)。

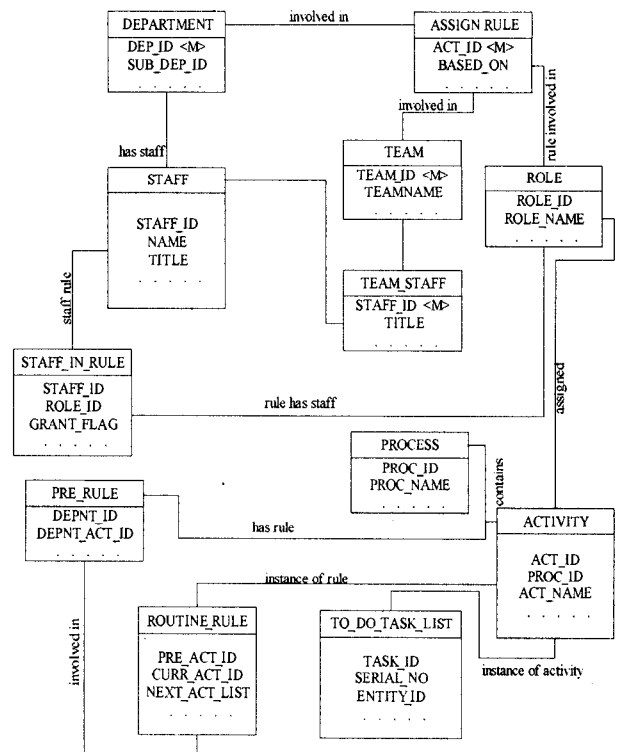


图 2 数据模型

(1) 组织模型。

与机构模型相关的表主要有 STAFF, DEPARTMENT, TEAM, STAFF_TEAM, ROLE 和 STAFF_IN_ROLE, 表之间的关系已经在图中通过不同含义的连线标出。下面将有重点地对其中的一些含义作出解释。

DEPARTMENT 和 TEAM 分别表示部门和功能组,部门通常表示纵向的行政隶属关系,而功能组通常表示横向的合作关系,也用来表示临时组织。DEPARTMENT 和 TEAM 分别通过相应的 UP_DEPT_ID 和 UP_TEAM_ID 关联使得在部门之间和功能组之间分别形成树状的上下级关系。STAFF 记录与人员相关的个体信息,其中的 LOGGING_ON 指示相应的人员目前是否已经登录到系统,ON_LEAVE 表示该人员是否正处于休假期,这两种信息在 workflow 引擎中将被作为任务分配的指示信息。STAFF 中的外码

DEP_ID 指明该人员所隶属的部门。关系 STAFF_TEAM 指定了人员与团队之间的隶属关系。

(2) 数据模型。

数据模型的核心是业务活动表(简称活动)ACTIVITY,其他相关的表结构主要有业务过程 PROCESS、业务规则(活动流转规则)ROUTING_RULE、活动前依赖规则 PRE_RULE、任务指派规则 ASSGN_RULE,任务列表 TO_DO_TASK_LIST 以及已完成的任务列表 HAVE_DONE_TASKS。

(3) 任务指派。

任务指派指的是依照何种准则将任务分配给具体人员来执行。只有常规交互活动才涉及到任务指派的问题;其他活动要么在前台不具备实际的应用逻辑,要么由 workflow 引擎自动调用,因此与任务指派没有任何关系。

在前文中已经提到了机构模型和数据模型中的许多表和字段将联合用于 workflow 引擎的任务指派,其核心表结构为 ASSGN_RULE,每一个常规交互活动在 ASSGN_RULE 表都对应一条记录。BASED_ON 指明任务指派的基准,它可以基于部门进行任务指派、基于团队进行任务指派、基于角色进行任务指派、基于自定义的方式进行任务指派,EXSE_FUNC 指明相应的自定义执行程序。

任务指派的基准确定了可以执行相应任务的群体,具体指派到哪些实际人员还取决于任务指派方法 METHOD。

2.2 workflow 引擎的控制模型

控制模型将机构模型和数据模型有机地结合在一起,它根据其中定义的业务规则对业务过程中的各项业务活动的流转以及任务指派等工作进行控制和协

调^[3]。控制模型是 workflow 引擎的控制中心。

引擎控制器是 workflow 引擎在运行时的控制中心,图 3 给出了引擎控制器的控制结构图。

(1) 调度中心。

调度中心接受从外部接口发送过来有关流程控制的请求(如业务初始化、获取任务以及结束任务等),然后根据不同的请求类型调用相应的处理模块完成与本次请求相关的操作并将结果返回^[4]。由于是在 DBMS 内部实现 workflow 引擎的控制模型,因此有关请求的并发处理等问题完全可以交给数据库管理系统来完成,也不需要诸如请求队列等形式的数据结构。因此,事实上可以将调度中心看成一个多线程的并发服务器,它可以对多个外部请求提供并发服务。在对外部请求的处理过程中肯定会涉及到对内部数据结构(即 workflow 引擎的数据模型)中有关数据的读写和更改操作,这些数据的完整性和互斥操作则可以通过 DBMS 提供的各种加锁机制来实现,从而实现了多个外部请求之间的独立性。

(2) 任务管理。

任务管理主要根据调度中心的指示完成诸如任务创建、任务状态的转换以及相关数据的维护等工作。每次“结束任务”的外部请求将触发调度中心调用“任务管理”为后继活动(如果存在的话)创建新的实例,其状态为“Pending”;同时,其他不同的外部请求也将触发“任务管理”实施任务状态的切换^[5]。任务状态转换图如图 4 所示。

(3) 任务指派。

任务指派处理只是针对常规交互活动,通常情况下,任务状态由“Pending”切换到“Waiting”过程中完成任务的指派工作,即处于就绪状态的任务在通常情

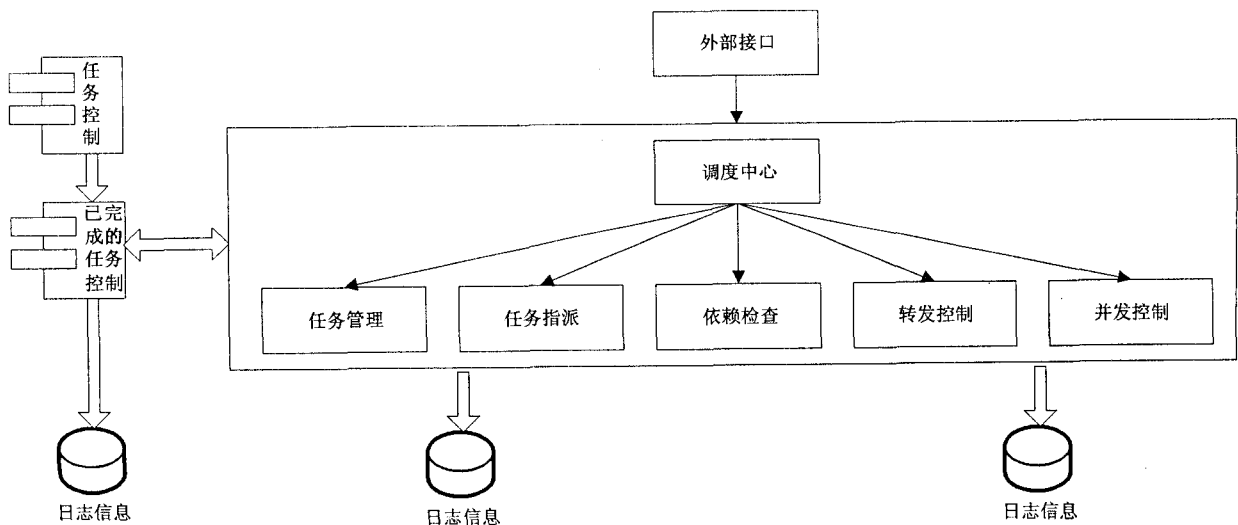


图 3 引擎控制器结构图

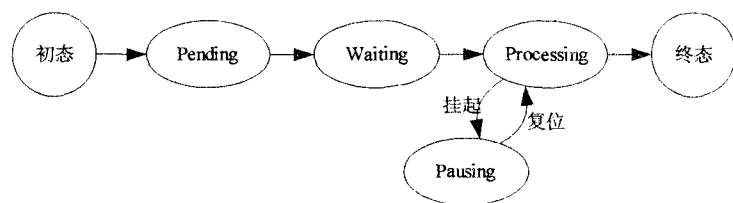


图 4 任务状态转换图

况下都确定了其执行者。任务指派过程首先根据任务指派基准确定可以执行此任务的群体人员,通常情况下这是一个包含多个人员的集合;然后根据任务指派方法确定由这个群体中的哪些个体来执行任务,执行任务的个体标识记录在相应任务记录的 STAFF_ID 字段中。

(4) 依赖检查。

依赖检查指的是活动的前依赖规则的检查,调度中心在将任务切换到就绪状态之前将进行相关的前依赖规则检查,只有满足检查条件的任务才可以进行状态的切换。前文已经描述了前依赖规则在数据模型中的表示方法,这里主要讨论在控制模型中是如何对各种前依赖规则进行处理的。

(5) 转发控制。

当应用发出“结束任务”的外部请求时,该请求将触发调度中心启动“转发控制”。转发控制的主要依据在工作流数据模型中定义的后转发规则,后转发规则定义了当前活动与其后继活动之间的关系。转发控制的处理过程是根据“结束任务”请求中所携带的“任务结束标记”以及相应前趋活动和当前活动的活动标识

匹配 ROUTING_RULE 表中的记录,从而得到相应的后继活动列表 NEXT_ACT_ID_LIST;然后由调度中心根据后继活动列表启动“任务管理”为相应的后继活动新建任务。

(6) 启动控制。

启动控制负责常规自动活动所对应的自动执行体的启动并对其活动进行监控。

3 结束语

文中的研究重点是在电子政务建设中引入基于 Web 服务的分布式 workflow 管理系统,探讨 workflow 模型的创建、分布式 workflow 框架的选择以及嵌入式 workflow 执行引擎的技术实现,进而发挥 Web 服务的优势来克服现有电子政务系统应用协调方面的不足。

参考文献:

- [1] 张晓东,柴跃廷,任守集.基于业务规则的事件驱动建模方法[J].清华大学学报,1999,39(7):25-28.
- [2] 李东波,徐平,韩祥兰.基于专家系统的工作流管理系统模型研究[J].南京理工大学学报,2001,25(1):96-99.
- [3] Sadiq W, Orłowska M. Analyzing process models using graph reduction techniques[J]. Information Systems, 2000, 25(2): 117-134.
- [4] 于蕾,王学通,王志晓.基于.NET技术的Web工作流模型研究[J].微机发展,2004,14(10):34-36.
- [5] 李宁,刘厚泉.一种支持分布式工作流模型的工作流管理系统[J].计算机应用研究,2006(3):205-213.

(上接第 42 页)

这样,就可以验证非高斯情况下,构造概率密度的函数表达式,其泛涵形式如下:

$$r(\bar{x}(t+1), \mathbf{K}(t+1), t)$$

可以调整增益矩阵 $\mathbf{K}(t+1)$,使之符合一个给定高斯形状的概率密度函数,这样即最坏情况的估计下,可以达到滤波的主观偏见最小的最优滤波估计。

3 实验结果分析及发展前景

由实验结果可知误差的方差具有最小值,此时增益非常接近高斯情况下计算出来的增益,因此滤波结果和常规方法算出的效果很接近,同时从宏观上对滤波器进行了最优的控制。通过熵的最小来设计非高斯滤波器的理论目前尚不成熟,但是它在非高斯滤波设计中却存在很广泛的应用价值,需要在应用中不断探索,总结经验。

参考文献:

- [1] Anderson B D O, More J B. Optimal Filtering[M]. New Jersey: Prentice-Hall, 1979.
- [2] 陆如华,何于班.卡尔曼滤波方法在天气预报中的应用[J].气象,1994,20(9):41-43.
- [3] 陆如华,徐传玉,张玲,等.卡尔曼滤波的初值计算方法及其应用[J].应用气象学报,1997(18):35-42.
- [4] 邓自立.最优估计理论及其应用——建模、滤波、信息融合估计[M].哈尔滨:哈尔滨工业大学出版社,2005:90-120.
- [5] 邓自立.最优滤波理论及其应用——现代时间序列分析方法[M].哈尔滨:哈尔滨工业大学出版社,2001.
- [6] 李彦鹏,黎湘,庄钊文.一种快速 kalman 滤波算法实现及效果评估[J].电子与信息学报,2005(1):153-154.
- [7] 周荫清.信息理论基础[M].第3版.北京:北京航空航天大学出版社,2002.
- [8] 王朝珠,秦化淑.最优控制理论[M].北京:科学出版社,2003.