

workflow 管理系统柔性机制

于 勇, 彭 岩

(首都师范大学信息工程学院, 北京 100037)

摘 要: 随着 workflow 管理系统在各个领域的应用, 系统的柔性机制引起人们的极大关注。该文介绍柔性化 workflow 管理系统的基本概念及架构, 描述从多个方面对 workflow 管理系统进行柔性化扩展的特点和实现机制。设计并实现一个具有更强适应能力的柔性 workflow 管理系统, 实践证明该机制有效。该系统的实现也为异地的业务流程处理, 甚至是跨平台间的业务组织和流程调用提供了良好的工作基础。

关键词: workflow; 柔性机制; workflow 基础

Flexible Mechanism of Workflow Management System

YU Yong, PENG Yan

(Institute of Information Engineering, Capital Normal University, Beijing 100037)

【Abstract】 Workflow management system applications are used in various fields now. Its flexible mechanisms cause great concern. This paper introduces the basic concept and structure of a flexible workflow management system. It describes the characteristics of the system's flexible expansion, and analyses those mechanisms to achieve it. A more flexible workflow management system with high adaptability is designed and developed. These mechanisms have been proved available with the practices. Implementation of the system is supplied for remote processing of workflow, and even for workflow's organizations and processing which is cross-platform.

【Key words】 workflow; flexible mechanism; workflow foundation

1 概述

workflow 是 workflow 的计算模型, 即将一个 workflow 中的业务或数据按照一定的逻辑和规则前后组织在一起, 并在计算机中以恰当的模型进行表示并对其进行计算或处理。workflow 管理系统是定义、创建、执行 workflow 的系统, 通过计算机技术的支持来定义、执行和管理 workflow, 协调 workflow 执行过程中工作之间以及群体成员之间的信息交互。由于信息技术的发展和日趋激烈的商业竞争, 人们不再满足于独立、零散的办公自动化和计算机应用, 而是需要综合的、集成化的解决方案, workflow 和 workflow 管理系统日益成为人们深入研究的对象。为了更好地适应不同领域流程处理的需求, 一个 workflow 管理系统应该具有较强的灵活性和适应能力以提高其业务处理的能力。本文就如何增强一个 workflow 管理系统的柔性进行了深入的探讨和研究, 并结合柔性化的思想在 .NET 平台下设计并实现了一个 workflow 管理系统。此系统功能完整、相较于传统 workflow 系统具有更强的适应能力。

2 workflow 管理系统的柔性化及其相关技术

2.1 workflow 管理系统

一个行业的业务处理的解决具有流程化的形式, 可以引入 workflow 技术规范业务处理的进行和提高业务处理的效率。一个规范的 workflow 管理系统^[1]主要包括 2 个部分:

(1) 流程定义工具, 也就是通常的流程设计器。用来创建计算机可处理的业务流程描述。这些描述可以是形式化的过程定义语言或对象关系模型, 也可以是简单地规定用户间信息传输的一组路由命令。

(2) workflow 执行子系统和 workflow 引擎: workflow 执行子系统也称为“workflow 运行环境”, 包括一个或多个 workflow 引擎。workflow 引擎是 workflow 系统的核心软件组元, 不同 workflow 引擎通

过协作共同执行 workflow。

2.2 柔性化

柔性化是指通过各种策略增强系统的适应能力, 使其能够更好地适应不同用户的使用需求。workflow 的柔性化主要指以下几个方面:

(1) 流程定义的柔性化^[2]。采用了通用性更强的 workflow 表示方法来设计和实现流程设计工具, 使其能够以更简单的形式体现用户设计的流程。这种不需要太多流程思想的设计工具可以适用于更多的普通用户。

(2) 流程运行的柔性化。为用户提供尽可能全面的流程运转模式, 包括自动化、半自动化的发散、聚合判断等。同时还应该为 workflow 的运转提供流程跳转和回滚机制, 使用户能够在必要的情况下更加灵活地对已运行的 workflow 进行控制。

2.3 WWF 技术^[3]

2006 年微软公司推出了其用于 Windows 平台下的 workflow 解决方案 WWF(Windows Workflow Foundation)开发框架, 专门用来在 .NET 开发平台下进行 workflow 管理系统的设计和开发。WWF 框架内部提供一个运行稳定、可信赖的 workflow 引擎, 并提供了一个完整的业务流程定义规范, 能够将各种类型的工作流按照一个统一的形式表示出来, 并稳定地运行于 workflow 引擎中。这种技术能够适用于各种行业和多种流程模式, 并且与 Web service 技术全面结合, 是目前 workflow 管理系统开发中流程表示和流程运行支持的最佳选择之一。

基金项目: 北京市属市管高校人才强教计划及北京市委组织部优秀人才培养基金资助项目(20061D0501600220)

作者简介: 于 勇(1979 -), 男, 硕士研究生, 主研方向: 人工智能; 彭 岩, 副教授、博士

收稿日期: 2008-04-11 **E-mail:** tayu2000@126.com

由于技术复杂, WWF 框架也存在一般工作流技术具有的流程设计难度大、缺少全局的“回滚”和“跳转”机制、对分布式处理支持不足等问题,在增强其柔性方面有着较大的研究空间。采用 WWF 技术使得人们可以把注意力集中在如何更好地提高工作流管理系统的柔性上,而不再需要考虑如何表示业务流程、如何建立流程的运行环境等问题。

3 柔性化工作流系统的实现

3.1 柔性化的系统架构

为了实现更为柔性化的工作流管理系统,结合 WWF 技术和“构件化”的设计思想,为系统引入了全新的架构。如图 1 所示,整个系统结构分为“基础构件层”、“单元模块层”和“工作流层”3 层。其中,“基础构件层”包括 WWF 提供的所有基础构件,负责向“单元模块层”提供可用的原始构件,其内容基本固定,只随着 WWF 本身的发展进行更新。“单元模块层”主要包括系统开发人员根据用户的需求使用 WWF 基础构件开发并封装后的单元模块,提供给流程设计工具和引擎使用。这些模块是根据需求设计开发的“业务逻辑”,具有原子性、一致性和持久性,但并未组织到工作流中。而“工作流层”用来负责所有流程的组织、运转和操作控制,用户使用的设计器、客户端和引擎等都处于这一层。

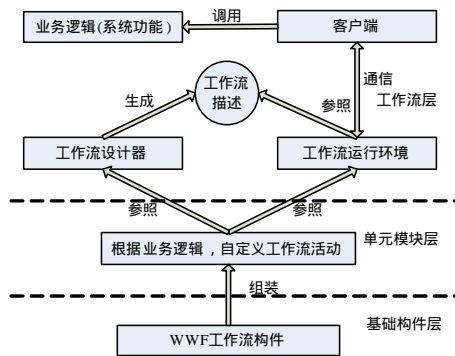


图 1 系统架构

与传统工作流管理系统相比,这样的架构具有以下优势:

(1)用户与复杂的流程设计技术相分离:用户只需要了解“单元模块层”上开发人员提供了哪些业务流程模块,然后进行简单的组织和设定即可,不再需要关注其深层次的实现方式。

(2)流程业务“逻辑”与“描述”相分离:定义一个“流程描述”来表示一个流程单元模块,工作流设计和运行控制时不再需要具体的“业务逻辑”,只在客户端运行时才会根据“描述”信息对相应的逻辑代码进行调用,客户端对业务的“逻辑”进行修改和升级不会对整个系统带来影响,如果“逻辑”的实现与 Web service 等技术相结合,也可以很容易地建立跨平台的流程调用。

(3)更强的扩展能力:当技术更新或用户需求发生变化时,只需要在相应的层次上添加新的模块或更新指定的模块,不需要对整个系统进行修改,也不会影响其他流程的运行。

(4)更好的应用前景:由于建立在统一的技术之上,这个架构可以很容易地适用于各种网络环境,也可以使用 C/S 和 B/S 2 种网络架构的系统共同完成一个工作流。

3.2 柔性化的流程定义工具

工作流设计工具是一个工作流管理系统的重要组成部分之一。由于流程设计较为复杂,传统工作流系统中用到的流程定义通常由系统开发人员来完成。当用户的需求发生变更

时,需要开发人员对使用的流程进行重新定义,降低了系统的通用性和扩展能力。实践证明,在当前许多工作流系统的实际应用中,如电子政务、人力资源管理、企业 ERP 系统等领域,用户往往更关注于流程运转的灵活性和系统使用的难易程度,而对系统是否能够支持工作流所有运转方式等并无严格要求。有别于传统的流程定义工具,为了增强系统的适应能力,本文主要研究如何在工作流的设计阶段更为简洁地表示一个复杂的流程、更有效率地进行工作流的设计。

3.2.1 流程表示

目前关于流程设计方面的研究集中在业务流程的表示形式上。包括 Petri 网、DAG 图(有向无环图)和 UML 活动图等对流程表示的优缺点和改进方法上。本文采用 DAG 图作为流程表示显示形式,并结合 UML 图表示的化简规则^[4],为用户提供了一个更为简单有效的流程表示方式。

3.2.2 流程转换

流程设计器能够根据设计好的流程图生成规范的 WWF 工作流描述。如图 2 所示,DAG 图作为流程表示的设计器中的流程图是一个有固定起点和终点的有向无环图,而 WWF 支持的工作流描述是一个 XML 字符串,也就是一颗 XML 树,其中流程的执行顺序和执行方式等由树分支的顺序和节点的类型来决定。如何将这个 DAG 图转换为 XML 树,并且既要符合用户的设计思路,又不能更改原设计的逻辑,这里需要一个严谨、完善的转换算法。

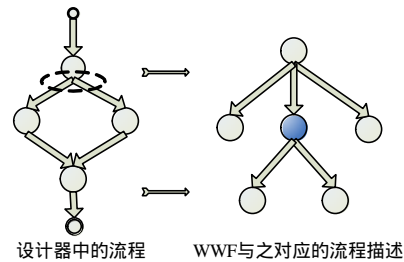


图 2 流程转换示意图

本设计器在进行转换设计时考虑了一般业务流程运转时可能遇到的各种分支类型,流程运行到当前活动时有多条后续活动可以选择时,设计者可以根据实际情况对系统提供的分支类型进行选择,包括:

- (1)并行分支:所有分支自动同步进行,当全部完成后重新聚合;
- (2)条件分支:根据用户定义的条件自动进行判断,选择一条分支运行;
- (3)手动分支:由用户手动激活后面的分支。

如图 2 所示,用户选择的分支类型信息会隐藏在 DAG 图的虚线处和 WWF 的有色节点(节点类型)上,系统进行流程转换时会根据用户的选择自动生成不同的 WWF 流程描述,从而工作流运行环境根据这些流程描述生成的工作流实例及其运行结果也会有相应的不同。

3.3 回滚和跳转机制的引入

3.3.1 “流程回滚”和“流程跳转”

流程的“回滚”和“跳转”是 2 个不同的概念,流程的“回滚”是指按照用户指定的要求返回到以前的执行状态,恢复执行过程中某一固定时间点或时间段的状态和数据信息。按用户控制方式来划分可分为“控制回滚”和“自动回滚”。系统在用户的控制下进行回滚可看作是控制回滚,而流程单元模块的原子性提供的“故障复位”功能就属于自动回

滚。流程回滚按回滚方式又可以分为“单步回滚”和“多步回滚”。

流程的“跳转”是指当有某些特殊需求或者特殊情况发生时,用户可能按照一定的策略跳过一些原计划应该执行的活动而直接进行后续的活动。流程的跳转具有以下特点:

(1)进行流程跳转后,“已执行活动”所产生的数据就可能失去意义,因此不能够再进行工作流的回滚,而使用这些数据进行的后续活动也可能得不到正确的处理结果;

(2)进行流程跳转后,若“当前活动”还有“后续活动”,则“工作流执行实例”仍然能够继续进行向后的跳转;

(3)流程终止也属于流程跳转的一种。

3.3.2 流程回滚的实现

系统中“流程回滚”是通过为工作流系统中的工作流运行环境添加持久化服务实现的。持久化服务是一个运行于工作流环境中的服务性程序,它能够自动在事先定义的时间点上将工作流运行的状态信息和数据保存到数据库中。这样就可以通过简单的“备份/还原”机制,在必要的时候将正在执行的工作流实例恢复到原有的状态。

在指定的时间点上将当前工作流实例的状态信息和数据保存到相应的数据表中。在用户确认了回滚操作后,系统将工作流运行实例的状态和数据信息恢复到指定的点上。通常定义的时间点是指在每执行完一个活动后,同样可以通过修改时间点的定义来控制“流程回滚”的粒度。

3.3.3 流程跳转的实现

流程跳转要结束当前执行的活动,直接开始执行后续的活动,也就需要动态地改变当前流程实例的流程定义,将后续指定的“活动”作为流程的“起始活动”开始执行。取得当前工作流实例的流程描述,并根据用户要求的跳转位置对其进行重新定义,删除掉“当前执行活动”和“指定执行活动”之间的工作项,然后结束当前活动,进入下一个活动。

4 关键部分的实现技术

4.1 工作流在 B/S 和 C/S 网络架构上的整合

当前工作流管理系统的开发和使用主要分为 B/S 和 C/S 2 种架构,与其他应用系统一样,两者各有其优势和适用范围。但由于产品较多和技术的不统一,目前的工作流管理系统在 2 种网络架构上的区分还很明显,往往一个流程在开发前就要确定其适用于哪种网络架构,且日后也难以修改。为增强系统的适应性,笔者在 B/S 和 C/S 的整合上进行了尝试。

本系统是在 .NET 平台基础上进行开发,.NET 平台不但提供了一个跨所有编程语言的统一的运行环境,同时其对类库使用方式^[5]的一致性也保证了流程在 2 种网络架构上都能很好地运行。本系统在开发了应用程序客户端(client)的同时,还针对一个简单的人力资源业务处理单元的具体业务操作进行了 Web 网站(browser)的开发,很好地利用了 WWF 在 2 种网络架构上的通用性。目前,用户能够通过浏览器或客户端同时对一个考勤管理的业务流程进行操作。在流程中,同一个业务能够使用客户端完成,也能够使用浏览器进行处理。

4.2 性能的优化和升级方式的改进

为了降低服务器的负载,工作流中大型数据的转移不宜全部放入工作流内部变量中。为此在数据库中建立 4 张表,用于保存状态信息和数据。包括“流程状态表”(保存工作流“当前”状态信息和内部变量)、“流程状态备份表”(保存流程每一步的状态信息和内部变量,当用户回滚时用于恢复),同时还添加了“大数据表”(用于保存工作流运行中暂存或输

出的大型数据,如文档、图像等)、“大数据备份表”(有条件保存每一步的大型数据,只有在数据发生变化时才予以更新,当 workflow 状态恢复时用于恢复内部数据)。这种带有冗余机制的数据处理不但能够保证系统对含有大型数据操作等业务处理的支持,同时能够更有效地提高系统运行的速度,维护数据的安全。

通常情况下,实际应用的工作流管理系统中升级最多的原因是具体业务逻辑发生了变化。如图 3 所示,在传统的工作流管理系统中,用户的操作只限于客户端的使用,一个流程的定义或业务逻辑发生变化时,设计器、客户端、服务器等都要由开发人员进行全面更新,为系统的升级带来了极大的不变。柔性化系统架构采用的“逻辑”与“描述”分离机制,使得开发人员只需要更新相应计算机上的相应模块即可,系统可以在不影响用户使用,甚至用户使用的同时进行升级。

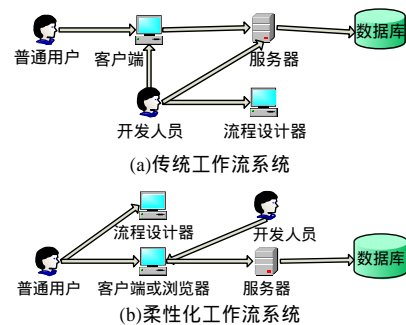


图 3 系统使用及升级对比示意图

5 结束语

目前,各个领域对于工作流的研究和应用越来越成为信息化建设的热点和重点。作为一种对常规性事务进行管理、集成的技术,工作流管理系统的出现是必然的,任何软件系统想要在其中使用工作流,也就需要建立或对外支持一个工作流管理系统。如何增强工作流管理系统的灵活性、适应性和可扩展能力是目前许多工作流管理系统面临的问题。

工作流管理系统的柔性化是扩展传统工作流管理系统能力的一个有益尝试。通过对工作流管理系统的柔性化改造,可以扩大工作流系统的使用群体,也可以使得业务流程本身的运转和控制更加灵活。本文的创新之处在于从多个角度对工作流管理系统柔性化的实现方式进行了深入分析,并开发了一个具有高度适应性的工作流管理系统。通过对实际使用的业务流程的实现研究成果进行了验证,也为后续的研究打下了良好的基础。

参考文献

- [1] Workflow Handbook[Z]. Lighthouse Point, FL, USA: Future Strategies Inc., 2004.
- [2] 王刚, 焦成柱, 章俊, 等. 基于 .net 图形化柔性工作流引擎的实现机制的研究[J]. 微计算机信息, 2006, 22(3): 11-12.
- [3] Box D, Shukla D. Simplify Development with the Declarative Model of Windows Workflow Foundation[EB/OL]. (2006-02-01). <http://msdn.microsoft.com/msdnmag/issues/06/01/WindowsWorkflowFoundation/>.
- [4] 汪文元, 沙基昌. 基于 UML 活动图化简方法的工作流模型校核研究[J]. 计算机工程, 2006, 32(15): 41-43.
- [5] 金正淑, 闫文耀, 陈亚军, 等. 基于 .NET 技术的网上办公模型研究[J]. 计算机工程, 2006, 32(12): 263-265.