

workflow管理系统的柔性技术研究与应用

申屠晓丽, 姚明海, 汤幸江

(浙江工业大学 信息工程学院, 浙江 杭州 310014)

摘 要:工作流的柔性是指灵活、动态地处理变更和异常的能力, 这种能力使得 workflow 管理系统能够满足企业业务流程不断变更的需求, 以及适应市场和企业的发展。自 20 世纪 90 年代中期开始, 在提高 workflow 管理系统的柔性方面进行了很多探索和研究, 引起了极大关注。该文首先介绍柔性 workflow 的相关概念及方法, 然后结合 workflow 可扩展模型和跳转返回机制来增强工作流的柔性, 最后基于微软 WWF(Windows Workflow Foundation)的开发平台, 通过示例来演示由宿主动态更新业务流程。

关键词: workflow; 柔性; 扩展

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2010)01-0120-04

Research and Application of Flexible Technology for Workflow Management System

SHENTU Xiao-li, YAO Ming-hai, TANG Xing-jiang

(Information Engineering College, Zhejiang University of Technology, Hangzhou 310014, China)

Abstract: Flexibility of workflow refers to the ability dealing with change and exception in a flexible and dynamic way. This ability enables workflow management system to satisfy demands of enterprise business process which keeps changing. How to improve the flexibility of workflow management system has aroused great concern since the mid 1990's. Focus on related concepts and methods of flexible workflow. It combines expansional model of workflow and implementation mechanism to enhance flexibility. Based on windows workflow foundation development platform, it finally shows how to dynamically update workflow by host process through demonstration.

Key words: workflow; flexibility; expansion

0 引 言

现代企业面临着快速多变、全球化的市场竞争环境, 为了提高企业的竞争力, 适应市场的需求和企业的发展, 需要不断调整和优化企业的业务流程。另外, 企业经营过程的执行逻辑在多数情况下并不是完全可以事先确定, 它需要根据流程的执行情况动态地发生变化。所以, 为了能够更加准确地表达企业的经营过程, 满足企业工作流程的实际需要, 提高 workflow 系统的柔性是十分必要的。

自 20 世纪 90 年代中期开始, 在提高 workflow 管理系统的柔性方面进行了很多探索和研究。柔性 workflow 主要体现在 workflow 的柔性建模技术和柔性执行技术,

旨在提供动态处理流程变更和适应异常情况的机制, 从而提高 workflow 管理系统的灵活性和动态性。这样, 开发灵活的 workflow 管理系统以支持企业动态的组织过程就成为当前 workflow 研究的重点。文中将探讨柔性 workflow 的相关概念和方法, 结合微软 WWF(Windows Workflow Foundation)开发架构, 深入研究柔性 workflow 的实现机制, 最后设计一个简单的动态更新 workflow 示例。

1 workflow管理系统的柔性研究及概念

根据国际 workflow 管理联盟(Workflow Management Coalition, WfMC)的定义, workflow 是“一类能够完全或者部分自动执行的经营过程, 它根据一系列过程规则、文档、信息或任务能够在不同的执行者之间进行传递与执行”。从定义中可以看出, workflow 是企业经营业务过程的计算机实现。workflow 管理系统则是完成 workflow 的定义和管理, 并按照在计算机中预先定义好的 workflow 逻辑推行 workflow 实例的执行的一个软件系统。工作

收稿日期: 2009-04-27; 修回日期: 2009-07-15

基金项目: 浙江省重大科技专项优先主题重点项目(2008C11SAB50014)

作者简介: 申屠晓丽(1983-), 女, 浙江东阳人, 硕士研究生, 主要从事企业集成信息化的研究; 姚明海, 博士, 教授, 研究方向为企业集成信息化和智能信息处理。

流管理系统的应用,帮助企业有效地改进和优化业务流程,提高业务处理效率和顾客服务质量等。

1.1 workflow管理系统的柔性分析

workflow管理系统的柔性是指灵活、动态地处理变更和异常的能力^[1]。具体来说,是指系统在其所处的环境、条件或者过程状态发生改变时,系统是否能够快速响应和跟随这种变化的能力。柔性主要体现在以下几点^[2]:

1) workflow模型的柔性定义:workflow模型应该充分表达各种复杂、灵活的业务过程,并且模型的定义支持系统运行时的动态修改或扩展;

2) workflow的柔性执行:workflow管理系统,特别是workflow执行服务,具有柔性执行机制,能够依据workflow模型动态更改workflow运行实例数据,包括workflow节点属性、操作,workflow流向等;

3) 活动任务的动态分配和资源动态调用:柔性workflow应支持根据运行情况动态确定任务执行人,以及对workflow相关资源的优化调用;

4) 柔性workflow的验证和仿真。

以上四点从柔性workflow研究的不同方面探讨workflow管理系统的柔性实现技术。

1.2 相关研究

workflow管理系统的柔性研究是近年来的一个热点问题。在理论研究上,国内外文献提出了不少方法,如文献[3]提出了继保留迁移规则,这是一种预防性的处理方法;文献[4]提出一种基于ECA规则和活动组合的动态workflow模型,该模型无需加入新元素,就可以对不确定因素进行封装;文献[5]提出的基于统一建模语言和多色集合的workflow集成建模方法,实现了workflow过程定义阶段的形式化建模方法;文献[6]在文献[5]研究基础上,提出支持workflow运行阶段动态变更的实例迁移方法等。要使workflow能动态适应业务流程变化,难点就是workflow实例迁移问题,即如何将运行中的实例迁移到新的workflow模型中^[7,8],具体的迁移流程如图1所示。

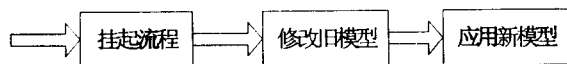


图1 workflow实例迁移流程图

流程实例有四种状态,实例化、执行中、挂起和结束,如图2所示。流程挂起就是将正在执行的workflow挂起,保留挂起时的状态信息,活动实例仍在运行但处于静止状态。在实际操作中,为了确保挂起的workflow能够顺利地过渡到新模型,在图1中的“修改旧模型”之前,需要区分当前运行区域和动态变更区域,然后考虑活动节点和流转路径的更改情况,再来选择相

应的迁移策略。该文接下来内容将利用流程的跳转返回机制,并结合WWF提供的工作流可扩展模型,在workflow执行过程中动态更改workflow逻辑,以适应业务流程变更的需求。

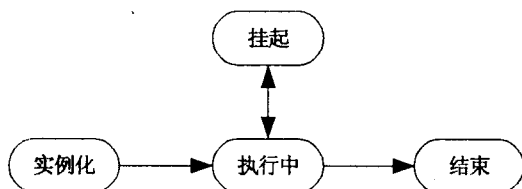


图2 流程实例状态转换图

2 柔性工作流的实现

当workflow管理系统运行某个workflow实例时,如果出现workflow实例的实际情况与在workflow模型定义时的不完全一致,就需要在workflow引擎的支撑下,扩展原先定义的活动。例如,动态添加活动情况,如图3所示,workflow引擎就要获得当前运行的workflow活动及其后续活动,然后将子流程(事先未定义的活动或流程)插入到当前活动的后面,子流程执行完后返回到调用流程,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,然后触发调用流程中的后继活动的执行,从而增强了workflow的柔性^[9]。

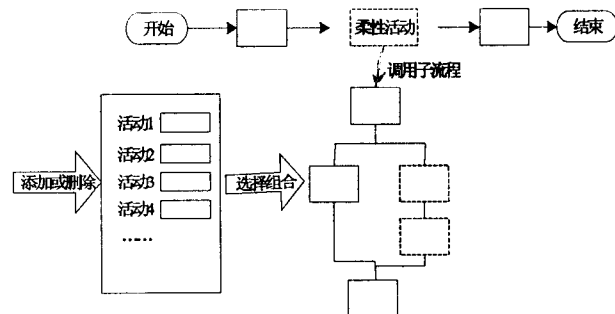


图3 柔性活动的执行过程图

2.1 流程的跳转返回

子流程执行完后,按用户指定要求返回到调用流程,恢复调用前的状态,正常的返回将触发调用流程中的后继活动的执行。当遇到某些特殊情况,流程的跳转返回还允许子流程在返回调用流程时,在约束范围内跳转到调用流程的后继活动之外的活动。

流程的跳转返回方式可分为单步和多步,具体有以下三种类型^[9]:

1) 正常跳转返回:子流程执行完后,返回触发调用流程的后继活动的执行。

2) 回退跳转返回:子流程执行完后,返回触发调用流程中子流程前面的某个活动执行。回退返回将形成一个循环,因此回退返回需要设定条件,当条件满足

时允许跳转,否则,按照正常跳转返回方式执行。

3) 快进跳转返回:子流程执行完后,返回触发调用流程中子流程后面的相距两个及两个以上的某个活动,快进有可能跨越一些活动,这些被跨越的活动相当于被删除不会被初始化和执行。

2.2 流程跳转返回的实现

工作流在整个生命周期中是透明且动态的,也就是可以对运行中的工作流实例进行修改。WWF 运行时服务提供的追踪、管理以及持久性服务是可扩展的。

流程的返回是通过 WWF 运行时服务提供的持久化服务和跟踪服务来实现。持久化服务是把未执行的工作流实例以及该实例中各种活动的状态存储在 XML 文件、数据库或者开发人员选择的其它方式,需要的时候将其加载回工作流运行时。跟踪服务是对工作流实例及其所包含的活动在运行时的状态进行跟踪,捕获并存储与工作流执行相关的数据到介质中,然后在分析时查询这些数据。就是通过这样简单的“备份/还原”,将正在执行的工作流实例恢复到原有的状态。

流程跳转需要取得当前工作流实例的流程描述,结束当前执行的活动,直接开始执行后续的活动,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^[10]。

2.3 WWF 技术

WWF(Windows Workflow Foundation)是微软推出的关于工作流的研发架构,是为微软产品及相关应用程序提供通用的工作流设计平台和开发工具。WWF 建立在 .NET 之上,是 .NET Framework 3.0 的重要组成部分。WWF 将复杂的工作流理论进行了封装,软件开发人员就可以在 Windows 平台上快速构建工作流应用程序、引擎和自定义活动。

Workflow 是一组 Activities,通过 Visual Studio 或者自定义设计器来定义工作流,Activities 通过工作流运行时 Runtime 来运行工作流和管理工作流的状态,通过运行时服务与工作流进行通信,最后整个工作流在宿主程序中被创建。WWF 框架是完全可扩展的,它提供了基本的活动以及一般的基本工作流功能,可以扩展这些活动以满足任何需求。在运行时期,可以查询工作流的状态和总体运行情况。WWF 提供的追踪特性也可以将日志信息记录到持久性介质上,甚至可以在运行时修改在这个平台上建立的工作流。这样,WWF 就提供了一个非常灵活的工作流的平台。

3 示例说明

以下考虑一个简单的客户订单审核流程。在该业

务流程中,当销售部门接收到客户订单时,业务员对订单进行初步审核,并把客户订单录入系统,内容包括:单据编号,日期,货品名称、规格、数量,交货日期,价格,客户等信息。然后根据客户资料信息进行信用审核,主要审查客户支付货款能力及信誉度。销售部经理根据信用审核的结果来决定是否与客户签订合同。但在实际业务执行时,发现在给客户发货过程中出现仓库缺货情况,耽搁合同的执行。为了保证合同顺利执行,不影响客户关系,就需要在原有的业务流程的“订单录入”后,除了对客户“信用审核”外,增加“库存检查”,根据实际情况下达生产任务,与客户进行及时有效协商,从而确保按时发货。整个过程如图 4 所示。

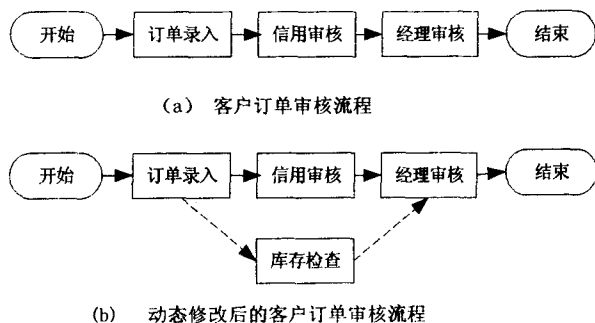


图 4 状态机工作流程图修改前后对比示意图

工作流程的动态变更主要表现在对活动和流程流转路径的修改上。为了在原流程基础上动态添加“库存检查”,采用 WWF 技术提供的持久性、追踪和动态修改功能,这样在开发过程中,就不必考虑如何表示业务流程、如何建立流程的运行环境等问题,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。

根据客户订单审核原流程,创建一个状态机工作流控制台程序,添加 4 个“state”活动,并建立状态间变迁关系。为了得到状态机工作流在运行中的状态变迁情况,启动 WWF 工作流的“Persistence”和“Tracking”服务,并将它们加载到工作流的运行时中,然后通过“StateMachineInstance. StateHistory”来获取状态机的状态。

为了保证订单顺利执行,在原流程上添加“库存检查”功能。在工作流开发中是通过动态添加 StateActivity 活动来实现的,即在宿主程序中添加一个按钮,并由此按钮的“Click”事件在状态机工作流运行时动态加载一个新的状态。

首先需要获取到整个工作流对象的定义,创建“WorkflowChanges”实例,再加载“State”活动并添加一个 EventDriven 子活动,代码如下:

```
Activity root = workflowinstance.GetWorkflowDefinition();
WorkflowChanges changes = new WorkflowChanges(root);
StateActivity storeCheck = new StateActivity();
```

storeCheck.Name = “库存检查”;

然后在“State”活动内添加 EventDriven 子活动, 代码如下:

```
EventDrivenActivity storeevent = new EventDrivenActivity();
```

```
storeevent.Name = “storeEvent”;
```

```
storeCheck.Activities.Add(storeevent);
```

最后, 建立状态之间的迁移关系, 通过“WorkflowChanges”对象应用到 workflow 实例中。

4 结束语

面对快速多变、全球化的市场竞争环境, 企业的业务流程需要不断调整和优化, 这对流程管理提出了更高的要求, 需要能够反映业务需求动态变化的并具有适应能力的柔性 workflow 管理系统。该文在研究 workflow 柔性特点和方法的基础上, 结合 workflow 可扩展模型, 在执行过程中动态更改 workflow 逻辑。然而, 该文是针对事先可预知但不确定的情况, 如何处理不可预见的异常情况^[11]和分布式 workflow 系统^[12], 将是今后工作研究和努力的主要方向。

参考文献:

- [1] 周建涛, 史美林. 柔性 workflow 技术研究的现状与趋势[J]. 计算机集成制造系统, 2005, 11(11): 1502-1507.
- [2] 王东勃, 王润孝. 基于多自主元的柔性 workflow 研究[J]. 计

算机集成制造系统, 2007, 13(5): 956-959.

- [3] Aalst W, Basten T. Inheritance of Workflows: An approach to tackling problems related to change[J]. Theoretical Computer Science, 2002, 270(1-2): 125-203.
- [4] 邓水光, 俞镇, 吴朝晖. 动态 workflow 建模方法的研究与设计[J]. 计算机集成制造系统, 2004, 10(6): 601-607.
- [5] 高新勤, 李宗斌. 基于统一建模语言和多色集合的并行设计过程研究[J]. 西安交通大学学报, 2006, 40(11): 1296-1299.
- [6] 高新勤, 李宗斌. workflow 管理系统动态变更模型与实例迁移方法研究[J]. 计算机集成制造系统, 2008, 14(8): 1521-1528.
- [7] 周明天, 王敏毅, 姚绍文. 一种基于扩展任务模型结构的工作流实例迁移方法[J]. 软件学报, 2003, 14(4): 757-763.
- [8] Yamaguchi S, Mishima A, Ge Q W, et al. A flexible and efficient workflow change type: selective shift [J]. IEICE, 2005, 88(6): 1487-1494.
- [9] 徐海军. 柔性 workflow 的研究[D]. 长沙: 中南大学, 2006.
- [10] 于勇, 彭岩. workflow 管理系统柔性机制[J]. 计算机工程, 2008, 34(24): 40-42.
- [11] Kammer P J, Bolcer G A, Taylor R N, et al. Techniques for supporting dynamic and adaptive workflow [J]. Computer Supported Cooperative Work, 2000, 9(3-4): 269-292.
- [12] 姜晨临, 蔡鸿明. 基于 Web 的分布式 workflow 系统的研究[J]. 计算机技术与发展, 2008, 18(6): 44-46.

(上接第 119 页)

```
0.0.0.0, 202.104.32.224, Network, 255.255.255.224, 22,
TYPE 1 EXTERNAL, 1, 202.104.44.34, 202.104.44.253
0.0.0.0, 202.104.32.192, Network, 255.255.255.224, 22,
TYPE 1 EXTERNAL, 1, 202.104.44.34, 202.104.44.253
0.0.0.0, 202.104.32.160, Network, 255.255.255.240, 22,
TYPE 1 EXTERNAL, 1, 202.104.44.34, 202.104.44.253
0.0.0.0, 202.104.40.0, Network, 255.255.255.0, 22, TYPE 1
EXTERNAL, 1, 202.104.44.34, 202.104.44.253
END-ROUTING-TABLE
```

通过分析仿真运行结果知系统仿真运行所得路由表与实际情况基本吻合。实验证明, 该系统设计合理, 并行性较好, 响应速度适中。

4 结束语

文中结合并行程序设计技术、分布对象技术、仿真技术详细介绍了虚拟计算机网络实验系统的设计和实现。实验结果表明, 该系统设计合理, 并行性较好, 响应速度适中, 拥有良好的开放性、通用性, 在网络教学和研究中均具有较好的应用价值。

参考文献:

- [1] Heybey A. Netsim Manual[D]. 美国: 麻省理工学院, 1989: 340-365.
- [2] 薛琪, 石冰心. 计算机网络仿真的数据处理[J]. 计算机仿真, 2000, 17(1): 49-52.
- [3] 唐宁九, 张旭. 分布式对象管理的几种常用技术[J]. 计算机应用研究, 2001(8): 74-77.
- [4] Milner R. Communication and Concurrency[M]. London: Prentice Hall, 1989.
- [5] 王晨, 周颖, 张德富. 一种并行分布对象的互操作模型[J]. 软件学报, 1999, 10(8): 861-867.
- [6] 齐宏, 冼建生. 计算机仿真、虚拟实验和网络实验室[J]. 中国远程教育, 2004(12): 71-73.
- [7] Meyer. Object-Oriented Software Construction[M] // 2nd ed. Professional Technical Reference. New York: Prentice Hall, 1999: 439-460.
- [8] 刘刚, 张德运, 李胜磊, 等. 一种虚拟路由器的体系结构及实现[J]. 计算机工程, 2004, 30(11): 87-88.