

基于主动数据库的 workflow 管理系统

刘培江, 傅秀芬, 陈长瑶

(广东工业大学计算机学院, 广州 510090)

摘要: 高端 workflow 管理系统在应用中与数据库的频繁连接和数据交换成为其性能的瓶颈。该文提出一种新的基于主动数据库技术和 ECA 规则理论实现 workflow 管理系统的方法, 探讨了基于主动数据库实现 workflow 管理系统的参考模型, 与传统 workflow 参考模型比较证明了该参考模型的合理性与可行性。

关键字: 主动数据库; ECA 规则; workflow 参考模型

Workflow Management System Based on Active Database

LIU Pei-jiang, FU Xiu-fen, CHEN Chang-yao

(Department of Computer, Guangdong University of Technology, Guangzhou 510090)

【Abstract】 Most of advanced workflow management systems are kinds of applications based on the database system, and it has been an bottle neck in data connection and exchange with database. This paper proposes a new kind of fruition of workflow management system based on the technology of active database and the theory of Event Condition Action(ECA) rule. A preliminary exploration is made into the reference model of implementation of workflow management system based on active database. Its reasonableness and feasibility are confirmed compared with traditional workflow reference model.

【Key words】 active database; ECA rule; workflow reference model

workflow 管理系统(WorkFlow Management System, WFMS)已经广泛应用于日常办公^[1], 一个有效的 workflow 管理系统可以提高工作效率, 促进人们的交流与合作。目前高端的 WFMS 大多基于数据库, 即所有的数据都存放在某种 DBMS 中, 系统的执行就是对数据库中数据的查询与操作^[2]。当用户量多、数据量大时, 难以保证其效率和性能^[3]。这是因为传统的数据库系统是被动的, 只能根据用户或程序的要求进行操作, 每次操作都需要与数据库进行连接和数据交换, 耗费大量资源。主动数据库技术能解决上述问题, 它可根据当前数据库的状态, 主动、快速、有效地进行相应处理, 不需用户或程序的干预。

1 主动数据库及其 ECA 规则

主动数据库系统是在传统数据库系统的基础上, 添加一个事件驱动的 ECA(Event-Condition-Action)规则库和事件监视器来实现的^[4]。ECA 规则库的建设是在主动数据库基础上实现 workflow 管理系统的核心。ECA 规则的形式化描述如下:

```
Rule Rule_name([Parameters]) ON (Events) {
  IF(condition 1)
  THEN actions 1
  [WHERE(constraint 1)]
  [EXCEPTION(exception handlings)]
  IF(condition 2)
  THEN actions 2
  [WHERE(constraint 2)]
  [EXCEPTION(exception handlings)]
  ...
  IF(condition n)
  THEN actions n
  [WHERE(constraint n)]
  [EXCEPTION(exception handlings)]
}
```

} /*“[]”内的内容为可选*/

一条 ECA 规则的最后是执行某个动作, 这个动作的执行又可能接连(cascade)^[5]触发其他 ECA 规则的执行, 形成密切联系在一起的 ECA 规则集。一个有效的规则集应具有的性质, 如终止性(termination)、汇流性(confluence)^[6]。

事件(event)是在数据库系统中某特定时刻对系统有意义的“发生”。基本事件通过逻辑操作符和一些专用事件操作符连接起来构成复合事件。常用的基本事件操作符见表 1。

表 1 常用的基本事件操作符

运算名称	运算符	用法	意义
并运算	AND	$e1 \text{ AND } e2$	$e1$ 和 $e2$ 同时发生的事件
或运算	OR	$e1 \text{ OR } e2$	$e1$ 和 $e2$ 至少有一个发生的事件
非运算	NOT	NOT e	e 不发生的事件
相继发生运算	·	$e1 \cdot e2$	$e1$ 结束后马上发生 $e2$ 的事件
之前发生运算	Before()	Before(e)	e 开始时就已经终止的事件
之后发生运算	After()	After(e)	e 终止时开始的事件
任意事件发生	Any()	Any($m, E1, E2, \dots, En$)	$E1, E2, \dots, En$ 中任意 m 个事件发生的事件
计数算子	Count()	Count(e, n)	事件 e 发生的次数为 n 时发生的事件

2 工作流业务在 ECA 规则上的表示

工作流是一种自动业务处理过程, 目的是实现或促使实现某个预期的业务目标, 按照某种预先定义的规则, 在多个参与者之间自动传递文档、信息或任务等^[7]。

工作流定义的核心是“业务”概念, 企业业务流程可看作是由不同的人参与的基本活动序列组成的。基本活动具有参与人员、时间等属性, 受资源条件的限制。

一个基本活动的定义如下:

Action=(Manipulation (parameters), ACB, AC, Exception-handl-

作者简介: 刘培江(1979—), 男, 硕士研究生, 主研方向: CSCW, 主动数据库, 支持向量机; 傅秀芬, 教授; 陈长瑶, 硕士研究生
收稿日期: 2007-03-25 **E-mail:** apeil100@sina.com

ing)

其中, Manipulation(parameters)描述基本活动执行处理的细节,包括处理的步骤、时间、输入输出等;ACB(Action Control Block)记录基本活动的属性信息,包括活动标志、所属业务的标志、参与人员属性、活动状态等;AC(Action Condition)说明进行基本活动的资源条件;Exception-handling 说明活动处理失败时要进行的操作。

业务的形式化定义如下:

Business=(ID, Actions, Rules)

其中, ID 表示业务的标志号; Actions 表示业务中所有基本活动的集合; Rules 表示进行基本活动的流程控制规则集合。

在主动数据库中,基本活动中 ACB 的信息可记录在一般的数据表中; Manipulation 是对这些数据表中数据信息的操作序列,对应主动数据库 ECA 规则的 Event 和 Action; Rules 可用主动数据库的 ECA 规则进行描述。

下面以一个经费审批业务为例,说明 workflow 业务在主动数据库 ECA 规则上的表示。图 1 是经费审批业务的基本活动流程,基本活动有:填写申请单(A1),部门经理审批(A2),总经理和财务总监审批(A3),财务部门办理(A4),结束流程并存档(A5),放弃申请(A6)。

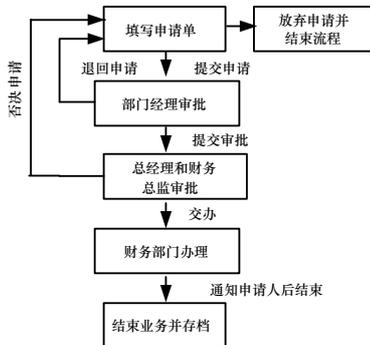


图 1 经费审批业务的活动流程

经费审批业务的形式化表示如下:

Fee-App=(fee-id, {A1,A2,A3,A4,A5,A6}, fee-rules)

其中, fee-rules 用 ECA 规则表示如下:

fee-rules= {rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9,rule10}

rule1: Rule rule1 () ON (Start(fee-id)) {
IF (Null) THEN Initial A1 }

rule2: Rule rule2 () ON (commit(A1)) {
IF (Null) THEN Initial A2 }

rule3: Rule rule3 () ON (drop(A1)) {
IF (Null) THEN Complete fee-id }

rule4: Rule rule4 () ON (A2) {
IF (Null) THEN Complete A1 }

rule5: Rule rule5 () ON (After(A2)) {
IF (Check-Result="yes") THEN Initial A3
IF (Check-Result="no") THEN Backto A1 }

rule6: Rule rule6 () ON (A3 or A1) {
IF (Check-Result="yes" or Check-Result="no") THEN Complete A2 }

rule7: Rule rule7 () ON (After(A3)) {
IF (Pass-Result="yes") THEN Initial A4
IF (Pass-Result="no") THEN Backto A1 }

rule8: Rule rule8 () ON (A4 or A1) {

IF (Pass-Result="yes" or Pass -Result="no") THEN Complete A3 }

rule9: Rule rule9 () ON (After(A4)) {
IF (Give-Notice="yes") THEN Complete fee-id }

rule10: Rule rule10 () ON (Before(Complete fee-id)) {
IF (Give-Notice="yes")
THEN Complete A4 And Put-on-file
IF (Give-Notice<>"yes")
THEN Complete A1//表示是由放弃申请引起的结束流程事件 }

3 基于主动数据库的 workflow 管理系统参考模型

本文提出一种新的基于主动数据库实现 workflow 管理系统的参考模型,如图 2 所示。

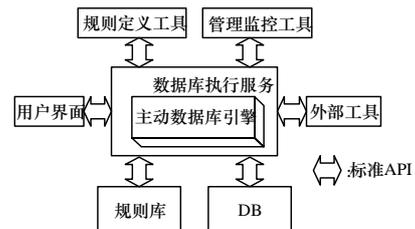


图 2 基于主动数据库的 WFMS 的实现模式

图 2 中实现模式的各部分组成及其功能概括如下:(1)数据库执行服务由一个或几个主动数据库引擎组成,负责获取用户的操作信息并进行处理、返回用户需要的信息、解释规则定义、维护规则库、管理传统数据库的数据、处理控制数据以及调用各种外部工具等工作,为 workflow 管理系统的定义、执行、监控提供一个运行时环境。(2)用户界面是用户进行操作与获取信息的窗口。(3)规则定义工具为用户提供定义 workflow 业务规则的操作环境。考虑方便性与直观性,其定义的内容由数据库执行服务并解释后写入规则库中。(4)管理监控工具负责管理并监控整个系统的运行情况,可对数据库中的 workflow 实例进行控制,并进行用户管理、数据统计等。(5)外部工具是辅助用户更好完成工作的功能集合,它可被数据库执行服务调用。(6)标准 API 接口为所有模块间的连接提供统一标准的服务,例如连接控制、数据格式转换、数据校验、获取和记录系统信息等功能。

传统 workflow 参考模型指 workflow 管理联盟于 1994 年 11 月底发布的工作流系统参考模型(workflow reference model),该模型详细描述了 workflow 系统的相关概念,并在此基础上给出 workflow 管理系统的各主要组成部分、各部分的功能及相互之间的接口^[2,7],如图 3 所示。

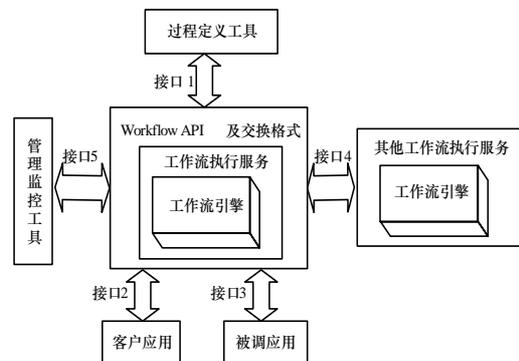


图 3 工作流参考模型

图 3 中参考模型的各个部分及其接口的功能如下:(1)过

程定义工具为用户提供对业务流程进行分析建模的手段,并生成能为计算机识别和处理的形式化描述。(2) workflow 执行服务由一个或几个 workflow 引擎组成,负责过程定义的解释、控制条件的计算、分派角色、维护 workflow 的控制数据等,完成 workflow 实例的创建、执行、管理与终止等,为 workflow 的执行提供一个运行时环境。(3) workflow 客户应用为用户提供一个工作项列表。(4) 被调应用是一些由 workflow 执行服务调用的功能应用的集合。(5) 管理监控工具负责监控并管理 WFMS 中 workflow 实例的状态。(6) 接口 1 为过程定义信息提供标准的交换格式和 WAPI 调用;接口 2 为 workflow 提供的各种服务,比如过程状态、流程控制、工作项列表维护及 workflow 实例管理;接口 3 提供一些标准的服务,基于这些服务可调用或开发一些专门的应用;接口 4 实现 WFMS 间的互连,定义 WFMS 间的互连模型、互连一致性级别和操作元素集等;接口 5 规定了在 workflow 执行过程中需要捕获和记录的各种信息,包括实例创建信息、工作项信息及远程操作信息等。

图 3 中模块和接口在新参考模型中的实现如表 2 所示。

表 2 2 个参考模型实现方式的比较

传统参考模型的模块及接口	在新参考模型中的实现办法
过程定义工具	主动数据库的 ECA 规则定义工具
workflow 执行服务	事件监视器、ECA 规则库及其规则管理器
workflow 客户应用	主动数据库的客户应用
被调应用	主动数据库的外部工具
管理监控工具	ECA 规则库及其规则管理器
接口 1	ECA 规则语义本身包含这个功能
接口 2	与主动数据库的客户应用的接口
接口 3	与主动数据库外部工具的接口
接口 4、接口 5	不需要,因为 ECA 规则库本身包含这些功能

由表 2 可知,传统参考模型的模块和接口在新参考模型

(上接第 44 页)

$v()$ 为 aspect 的结果与 bef-call 重构方式相似,区别在于重构前 $u(ups)$ 位于 $v(vps)$ 之前,重构后 aspect B 的通知变为 after。

(6) aro-aft-exec 重构方式。aro-aft-exec 重构方式重构横切关注点 $v()$ 为 aspect 的结果与 aro-bef-exec 重构方式相似,区别在于重构前 if (cond) { $v(vps)$;} 位于 ss 之后,重构后 aspect B 中 proceed() 位于 if (cond) { $v(vps)$;} 之前。

3 应用示例

为了评估面向方面的自动化重构方法,笔者将该方法应用于“基于 workflow 机制的 MIS 系统开发工具”中,实现该工具面向方面的自动化重构。“基于 workflow 机制的 MIS 系统开发工具”采用 Java 语言编写,包含大约 13 000 行代码和 93 个类。

自动化重构过程中,识别横切关注点 47 处,其中相同调用模式的横切关注点 31 处,不同调用模式的横切关注点 16 处,Statement Reordering 重构发生 3 次,占全部横切关注点的 6.4%。由此看出,大部分横切关注点(93.6%)可直接确定重构方式。47 处横切关注点确定的重构方式如表 1 所示。

表 1 重构方式统计

重构方式	发生次数	所占比例(%)
bef-call	13	27.7
aft-call	7	14.9
bef-exec	17	36.2
aft-exec	8	17.0
aro-bef-exec	2	4.2
aro-aft-exec	0	0.0

中都有实现。传统方法设计 workflow 管理系统时须考虑参考模型中所有模块的设计问题,而在主动数据库中很多问题都不考虑,增加了 workflow 管理系统标准化工作的可行性。

4 结束语

本文方法发挥了数据库组织管理数据的优势,能避免传统 workflow 管理系统面临的瓶颈问题,是对主动数据库技术与 workflow 技术的结合的新尝试。主动数据库技术应用的参考模型,不仅适用于实现 workflow 管理系统,也为其他基于主动数据库技术的应用提供了参考。

参考文献

- [1] 曹健,张申生,黄鹤远,等.基于 ECA 规则的适应性 workflow 技术研究[J].计算机集成制造系统——CIMS,2002,8(9):737-741.
- [2] 史美林,向勇,杨光信,等.计算机支持的协同工作理论与应用[M].北京:电子工业出版社,2000.
- [3] 李建中.日新月异的数据研究领域——数据库技术的回顾与展望[J].计算机应用与软件,2003,20(11):13-18,50.
- [4] 汤庸,叶小平,汤娜.数据库理论及应用基础[M].北京:清华大学出版社,2004.
- [5] Campin J, Paton N. Specifying Active Database System in an Object-oriented Framework[J]. International Journal of Software Engineering and Knowledge Engineering, 1997, 7(1): 101-123.
- [6] Aiken A, Hellerstein J M. Static Analysis Techniques for Predicting the Behavior of Active Database Rules[J]. ACM Transactions on Database Systems, 1995, 20(1): 3-41.
- [7] Workflow Management Coalition. The Workflow Reference Model [EB/OL]. (2006-02-20). <http://www.wfmc.org/standards/docs/tc003-v11.pdf>.

由表 1 可以看出, aro-bef-exec 和 aro-aft-exec 重构方式占重构总数的 4.2%,这两种重构方式相对不重要。

“基于 workflow 机制的 MIS 系统开发工具”的面向方面的自动化重构结果表明,针对方法调用,本文提供的 6 种重构方式已经足够。

4 结束语

本文提出了一种将面向对象程序自动重构为 AOP 的方法。首先对面向对象源程序进行横切关注点识别,然后确定横切关注点的重构方式,最后执行重构。使用该方法实现了“基于 workflow 机制的 MIS 系统开发工具”的 AOP 移植。面向方面的自动化重构方法仍需大量的研究工作,如自动化重构过程中形成的易碎切入点问题。

参考文献

- [1] Elrad T, Filman R E, Bader A. Aspect-Oriented Programming[J]. Communication of the ACM, 2001,44(10): 29-32.
- [2] Marin M. Refactoring JHotDraw's Undo Concern to AspectJ[C]//Proc. of the 1st Workshop on Aspect Reverse Engineering. Washington D.C.: [s. n.], 2004.
- [3] Monteiro M. Refactorings to Evolve Object-oriented Systems with Aspect-oriented Concepts[D]. Portugal: Universidade do Minho, 2005.
- [4] Cole L, Borba P. Deriving Refactorings for AspectJ[C]//Proceedings of AOSD'05. [S. l.]: ACM Press, 2005.
- [5] Braem M, Gybels K, Kellens A. Automated Pattern-based Pointcut Generation[C]//Proceedings of SC'06. Vienna, Austria: [s. n.], 2006.