

workflow 迁移问题的研究

檀迎军¹, 侯秀萍¹, 赵云峰²

(1. 长春工业大学 计算机科学与工程学院, 吉林 长春 130012;

2. 装甲兵技术学院 控制工程系, 吉林 长春 130117)

摘要: 为了提高 workflow 对动态不确定因素的处理能力和解决运行中 workflow 流程发生变化的问题, 重点对 workflow 的迁移问题进行了研究。为了使运行中的 workflow 能够向新 workflow 顺利迁移, 以满足企业的实际应用, 在动态 workflow 模型的基础上, 提出了一种基于事件-条件-活动规则和活动选取规则的动态 workflow 模型的迁移算法。通过一系列的规则, 使运行中的 workflow 在保证执行路径的查询完整性和未执行活动的业务完整性的基础上能够顺利迁移到新的 workflow 中, 以解决企业实际运行中流程发生变化的问题。

关键词: 迁移; 重组; 完整性; 动态 workflow

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)01-0139-04

Study of Workflow Transfer Problem

TAN Ying-jun¹, HOU Xiu-ping¹, ZHAO Yun-feng²

(1. College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China;

2. Department of Controlling Engineering, Armor Technique Institute of PLA, Changchun 130117, China)

Abstract: Study the workflow transfer problem in order to improve the workflow's disposal ability to the factor of dynamical and uncertainty and solve the problem of the workflow's change while on running. In order to make the running workflow can migrate successfully to new workflow to satisfy the practical application of the enterprise, based on the model of dynamic workflow, bring forward a transfer arithmetic of dynamic workflow model which based on affair-condition-activity regulations and activity selection regulation. Through a series of regulations, it makes the workflow in running can migrate to new workflow successfully on guarantee of the query integrality of the execution path and the business integrality of the activity unexecuted to solve the problem of workflow's change during enterprise practical running.

Key words: transfer; recombine; integrality; dynamic workflow

0 引言

在企业信息系统的建设中, workflow 技术受到越来越多的重视。随着企业环境的持续变化, 不确定性和多变性已成为现在企业流程的内在特点, 但大多数 workflow 软件产品仅对可预见、可事先给出完整定义的流程进行管理, 对流程动态不确定因素缺乏支持^[1]。所以提高 workflow 系统应对流程变化的能力, 已经成为 workflow 领域研究的热点问题之一。根据业务流程变化的特点, workflow 被划分为柔性 workflow、自适应 workflow 和动态 workflow^[2]。其中柔性 workflow 在业务流程发生变化的情况下能够做出相应的改变, 其难点在于如何将运行

中的 workflow 实例迁移到新的 workflow 模型中^[3~5]; 动态 workflow 在模型定义不完整的情况下能生成流程实例并运行, 其难点在于如何对部分流程信息进行建模。实际的工作流往往都有一个很长的时间跨度, 在运行过程中难免会遇到一些突发事件如部门的变动、人员的调动和流程的变化等, 为了提高 workflow 对动态不确定因素的处理能力以及增加对运行中的 workflow 业务流程发生变化的处理能力, 文中从一汽启明公司 workflow 产品的实际应用入手, 结合柔性 workflow 和动态 workflow 的优点和特点, 提出了一种基于 ECA(Event-Condition-Action) 规则和活动迁移规则的动态建模机制, 该机制在动态建模的基础上主要研究如何将运行中的 workflow 迁移到新的 workflow 中, 文中引入了一套规则机制来约束活动的迁移(添加活动、删除活动、更新活动状态), 从而实现了把运行中的 workflow 迁移到新的 workflow 中继续执行。

收稿日期: 2008-04-30

基金项目: 吉林省科技发展计划重大科技攻关项目(20040305)

作者简介: 檀迎军(1981-), 男, 河北邢台人, 硕士研究生, 研究方向为软件工程; 侯秀萍, 教授, 研究方向为软件工程。

1 动态 workflow 定义

workflow 模型是通过定义任务(活动)、任务间逻辑顺序、数据和资源对流程进行抽象处理的,在动态 workflow 模型中,把 workflow 中的活动分为三种:一种是静态活动(static activity),此类活动是事先可预见的活动,能给出具体定义,主要指开始活动和结束活动;一种是人为活动(artificial activity),此类活动是人参与执行的;第三种称为机器活动(machine activity),此类活动是机器自动执行的,不需要人工的参与,主要负责路由选择、条件判断、状态赋值和规则的执行等。

在迁移算法的研究中,把迁移策略分为三种:

①软迁移:按照事先制定的规则,只迁移允许迁移的节点,优点是能使流程继续执行,缺点是对于已经执行的节点无法干预;

②缓迁移(针对迁移):符合软迁移规则的节点做迁移,不符合软迁移规则但又发生改变的节点做迁移标志(更新、添加、删除),在此流程下次重新执行前先做一次重组,再执行,优点是可针对一个流程做迁移,不用定制新的模板,能使流程继续执行,缺点是对于已经执行的节点无法干预;

③硬迁移:不管流程的运行状态和运行结果,对所有的节点做迁移,迁移结束后重新执行流程,优点是能够使所有的节点都能够按预先的设想执行,缺点是无法消除所有已执行节点带来的影响,系统开销大。这里重点研究软迁移的迁移策略,其它策略及实现以后再做介绍。

workflow 的相关定义如下:

定义一: $WFL = \langle ID, M, A, E, VC, STATUS, FLAG \rangle$ 表示一个完整的动态 workflow,其中 ID 为活动的唯一标识;M 为 workflow 的基本信息集合;A 为 workflow 的活动集合;E 表示 workflow ECA(Event - Condition - Action)规则集合,一条 ECA 规则由事件(event)、条件(condition)和行为(action)三个元素组成,表示当一个 event 发生时,若 condition 满足,则 action 被触发^[6];STATUS 表示 workflow 的迁移状态, $STATUS \in \{N_move, Y_move\}$, N_move 和 Y_move, 分别表示运行过程中不做迁移、运行过程中随新 workflow 的产生而迁移;FLAG 为 workflow 的迁移标志位, $FLAG \in \{ID, NULL\}$, workflow 做迁移后,把 FLAG 的值改为新 workflow 的 ID,反之为 NULL,当 workflow 下次启动时,如果 FLAG 不等于 NULL,此次启动时直接引用新的 workflow,其模板号为 ID,VC 表示 workflow 变量的集合。

定义二: $M = \langle NAME, C_person, C_time, M_time \rangle$ 表示 workflow 的信息集合,workflow 的名称 NAME,创建者 C_person,创建时间 C_time,修改时间

M_time。

定义三: $a_i = \langle name_i, type_i, message_i, AS_i, status_i, location_i, images_i \rangle$ 表示 workflow 中的一个活动,其中 $name_i$ 为该活动的名称; $type_i$ 为该活动的类型, $type_i \in \{start, man, machine, end\}$,当 $type_i = machine$ 时 AS_i 中的 SR 不为空,反之 AS_i 中的 SR 为空; $message_i$ 为该活动的信息集合; $status_i$ 为当前活动的状态, $status_i \in \{no - startup, executing, executed\}$,当 $type_i = machine$ 时, $status_i \in \{no - startup, executed\}$,当 $type_i \in \{start, end\}$ 时, $status_i = null$,因为 start 和 end 只是一个用来标示开始和结束的标记; AS_i 为动态活动具体化时的选取规则库, $location_i$ 表示活动在工作流程图中的坐标, $images_i$ 表示活动的图形化标识。

定义四: $AS_i = \{AC, SR\}$, AS_i 表示动态活动具体化时的选取规则库,为一个二元组:

① $AC = \{a_1, a_2, a_3, \dots, a_i, \dots, a_n\}$ 为动态活动具体化时可选取的活动集合。

② SR(select rule) 为活动执行时的决策规则库。

定义五: VC(variable concurrence) workflow 变量集合:

$VC = \{vc_1, vc_2, vc_3 \dots vc_i, \dots vc_n\}$; 其中 $vc_i = \langle v_name, v_type, v_value \rangle$ 表示 workflow 的一个变量, v_name 为变量名, v_type 为变量类型, v_value 为变量值。

2 迁移算法 WMR

当 workflow 发生变化时,重新定制一个新的 workflow,对所有现行的流程做一次迁移,把变动节点增加到部分允许的工作流程中。即在迁移之前先制作一个迁移用的新 workflow 模板 G2,制作成功后扫描正在运行的所有 workflow G1,逐一进行迁移。

迁移过程满足的约束规则是:

①为保证执行路线的查询完整性,已执行的节点不做迁移操作;

②为保证 workflow 的业务完整性,正在执行的节点不做迁移操作;

③未启动的节点可以做更新迁移;

④两个未启动的节点之间可做添加新节点的操作;

⑤从未启动节点开始到已启动节点结束两者之间可以做新增路径的迁移;

⑥下次运行时,直接启用新的 workflow 模板。

算法描述如下:

G1:正在运行的 workflow, G2:要迁移到的新 workflow, G: 迁移后的新 workflow, 返回结果为迁移后的工作

流。

```
//move - arithmetic
for every activity  $a_i$  in G
{
    if  $a_i$  has moved//如果  $a_i$  已经做过迁移, 不做任何操作, 开始遍历下一个节点
        continue
    find the activity a in  $G_2$  where a is equipollence with  $a_i$ 
    if  $a_i$  is same with a //说明未做任何更新操作, 直接迁移
        move the activity  $a_i$  to G
    else if can not find the equipollence activity in  $G_2$ //说明该活动已被删除
    {
        if the status of  $a_i$  is executing or executed//执行中和已执行的不迁移
            move the activity  $a_i$  to G
        else if the status of the parent activity of  $a_i$  is executing
            move the activity  $a_i$  to G //父节点正在执行, 子节点不做删除迁移
        else//不迁移至目标 workflow G 即删除了该活动节点
            continue
    }
    else//说明  $a_i$  在新 workflow 中作了更新, 应该做更新迁移
    {
        instead of the AS of  $a_i$  by the AS of a
        move the activity  $a_i$  to G
        traversal the  $G_2$  from the activity a to the activity  $a_2$  which is the first activity exist in  $G_2$  and  $G_1$  and move every activity traversed to G //将遍历的所有节点都迁移到 G 中
    }
}
```

3 workflow 正确性检测算法

从 workflow 的开始节点开始遍历, 如果每个节点最终都能到达终点, 则称该 workflow 是合法的, 否则该 workflow 模板是不合法的, 即不允许存在不可到达的节点, 所有节点都应该至少存在于一条由开始节点到结束节点构成的路径中。算法描述如下:

```
//check the workflow
while(if the queue is not empty)//如果队列非空
{
    get the activity a from the queue;
    if a has sub Activity//如果 a 有子活动节点
    {
        get the first sub activity b from the concourse of sub activity of a;
        insert b to the queue;
        insert b to temp activity concourse;
        delete b from the concourse of sub activity of a;
    }
}
```

```
else//说明该节点 a 为终点、遍历过的节点或是孤立的不可到达终点的节点
{
    if the activity a is instance of end point//说明 a 是终点, 所以可到达 a 的节点都是合法节点
    {
        move the activity c to the right concourse from the temp concourse;
        clear the temp concourse;
    }
    else if the activity a in the right concourse //说明 a 是遍历过的可到达节点
    {
        move the activity c to the right concourse from the temp concourse;
        clear the temp concourse;
    }
    else
        return "The workflow template is lack of logic.";
}

if the queue is empty
    return "The workflow template is legality.";
else
    return "The workflow template is lack of logic.";
```

4 应用举例

在保险公司参加意外保险, 当发生意外时, 首先向保险公司提出索赔申请, 保险公司接到索赔申请后, 对索赔条目进行分析和核实, 当核实无误并认为在保险范围内时开始立案, 此时保险公司会估计赔款金额, 最后将赔款支付给索赔者, 据此制定 workflow 并取一运行中的实例, 此实例已运行至进行立案节点, 如图 1 所示。

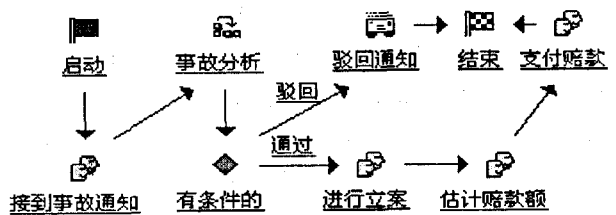


图 1 运行中的 workflow

数据结构如下:

EFL = < ID, M, A, E, VC, "Y_move", NULL >

- ① ID = "No. 1"
- ② M = ("保险索赔", "檀迎军", "2008 - 04 - 03", "")
- ③ A = {a, b, c, d, e, f, g, h}

a = (“启动”, “start”, “”, AS_a, “executed”, (0, 30), “start.gif”)

b = (“接到事故通知”, “man”, “”, AS_b, “executed”, (30, 30), “man.gif”)

c = (“有条件的”, “machine”, “”, AS_c, “executed”, (90, 30), “select.gif”)

d = (“进行立案”, “man”, “”, AS_d, “executing”, (120, 30), “man.gif”)

e = (“估计赔款数额”, “man”, “”, AS_e, “no - startup”, (150, 30), “man.gif”)

f = (“支付赔款”, “man”, “”, AS_f, “no - startup”, (180, 30), “man.gif”)

g = (“发驳回通知至当事人”, “man”, “”, AS_g, “no - startup”, (90, 0), “email.gif”)

k = (“事故分析”, “machine”, “”, AS_k, “executed”, (60, 30), “select.gif”)

h = (“结束”, “end”, “”, AS_h, “no - startup”, (180, 30), “start.gif”)

AS_a = (AC, SR) AC = {b} SR = NULL ; AS_b = (AC, SR) AC = {k} SR = NULL

AS_c = (AC, SR) AC = {c} SR = {if (flag) turn to d; else turn to g;}

AS_d = (AC, SR) AC = {e} SR = NULL ; AS_e = (AC, SR) AC = {f} SR = NULL

AS_f = (AC, SR) AC = {h} SR = NULL ; AS_g = (AC, SR) AC = {h} SR = NULL

AS_k = (AC, SR) AC = {c} SR = NULL

④ VC = {vc1 | vc1 = (“flag”, “java.lang.Boolean”, “false”)}

但是随着实际的应用会发现,当发生意外事故时,当事人和保险公司都希望将损失降到最低,所以对保险公司来说,对于某些事故,要采取一些应急措施进行挽救,所以索赔立案后,在进行赔款额估计的同时还要采取一定的应急措施来降低双方的损失,当估计赔款数额和应急措施都进行完后,再次调整赔款额,最后向用户支付赔款,所以需要在原工作流中增加三个节点:应急措施、阈值和调整赔款,其中阈值充当运行中的监控,监控估计赔款额和应急措施是否都执行结束,如果两者都运行结束,才会触发下一节点(调整赔款额),如图 2 所示。

按照工作流的迁移规则,“有条件的”活动之前的所有活动都已执行结束,直接迁移,进行立案节点发生了改变,做更新迁移,从进行立案节点开始遍历,支付赔款节点为第一个同时存在于运行中和要迁移到的工

作流中的节点,将要迁移到的新工作流中从进行立案到支付赔款之间的节点直接迁移,支付赔款和结束节点也做更新操作(坐标发生了变化),遇到结束节点,迁移结束。按迁移算法执行迁移后,迁移后的 workflows 见图 2,发生变化的数据结构如下所示:

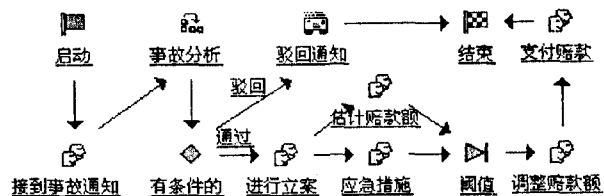


图 2 要迁移到的新工作流

EFL = < ID, M, A, E, VC, “Y - move”, “No. 2”

>

② M = (“保险索赔”, “檀迎军”, “2008 - 04 - 03”, “2008 - 04 - 22”)

③ A = {a, b, c, d, e, f, g, h, m, n, t}

d = (“进行立案”, “man”, “”, AS_d, “executing”, (120, 30), “man.gif”)

e = (“估计赔款数额”, “man”, “”, AS_e, “no - startup”, (150, 30), “man.gif”)

m = (“应急措施”, “man”, “”, AS_m, “no - startup”, (180, 40), “man.gif”)

n = (“阈值”, “machine”, “”, AS_n, “no - startup”, (210, 30), “check.gif”)

t = (“调整赔款额”, “man”, “”, AS_t, “no - startup”, (240, 30), “man.gif”)

AS_d = (AC_d, SR_d) AC_d = {e, m} SR_d = NULL

AS_e = (AC_e, SR_e) AC_e = {n} SR_e = NULL;

AS_m = (AC_m, SR_m) AC_m = {n} SR_m = NULL

AS_t = (AC_t, SR_t) AC_t = {f} SR_t = NULL;

AS_n = (can, SR_n) AC_n = {t}

SR_n = {If the status of e and the status of m are both executed Turn to t;}

按照工作流正确性检测算法对迁移后的工作流进行正确性检验,证明发生变化的部分仍为可到达节点,迁移后的工作流合法,到此整个工作流迁移结束。

5 结束语

在动态建模的基础上给出了迁移算法,在该算法中结合动态建模,给出了一种运行中的工作流向新工作流迁移的方法,即运行流程发生变化时的一种解决方法。这样一来,当工作流在企业实际运行中流程结构发生变化时就能够定制一个新的适应需求的新工作

(下转第 146 页)

同时间长度内,分别进行了多次的实验,对多次的实验取其平均值。运行结构如图 5 所示。图中,横坐标为时间,纵坐标为车辆到达目的地的百分比,即到达车辆数与发车数的比值。

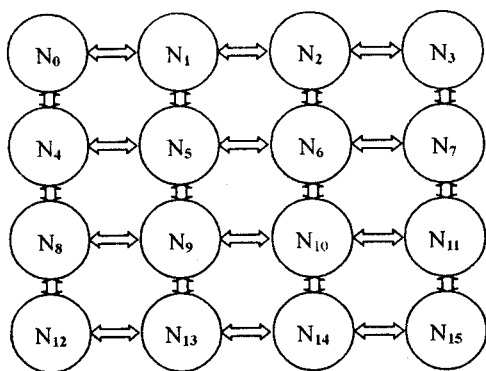


图 4 路网结构

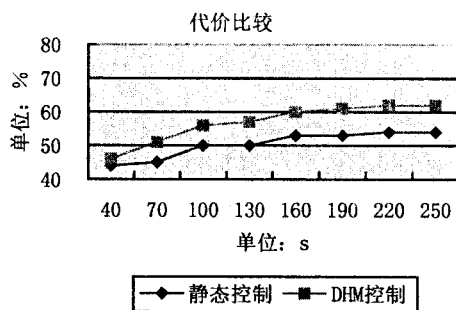


图 5 规则路网结构的代价比较

从实验的结果可以看到,采用 FDHM 对信号灯进行控制,都能使运行代价有很明显的改善。此外,从图中可以看到:系统运行时间越长,所改进的效果越明显。

5 结束语

文中介绍了目前对信号灯优化配时的研究。首先

阐述了信号灯配时优化方面的相关基础知识。然后详细介绍了数字荷尔蒙模型在信号灯配时中的应用,并结合模糊控制理论,提出了一种新的信号灯配时优化方法:基于数字荷尔蒙模型的模糊控制算法(FDHM)。最后,对该理论在信号灯控制方面的应用进行了仿真实验。智能交通系统研究的目的在于实现整个城市交通控制信号的全局优化,即单个交叉口的控制要服从全局的调度,局部的最优要服从全局的最优。数字荷尔蒙模型及其与其他算法的协调应用仍是今后研究的重点。

参考文献:

- [1] 丁爱萍,杜干.基于模糊控制的智能交通系统[J].微计算机信息,2006(13):30-31.
- [2] 曹洁,苏玉萍,吴国龙,等.城市交通信号灯两级模糊控制及仿真研究[J].交通与计算机,2007(1):82-85.
- [3] 李威武.城域智能交通系统中的控制与优化问题研究[D].杭州:浙江大学,2003.
- [4] 汤志康,郑积仕,王伟智.基于模糊神经网络的单交叉口变相序控制[J].浙江交通职业技术学院学报,2006(2):29-32.
- [5] 杨煜普,欧海涛.基于再励学习与遗传算法的交通信号自组织控制[J].自动化学报,2002(4):564-568.
- [6] 高丽颖,陈阳舟,李振龙.基于 Agent 控制器的单路口信号灯学习控制方法研究[J].交通与计算机,2007(5):59-62.
- [7] Shen Wei-Min, Will P, Galstyan A. Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms [D]. Marina del Rey, CA, USA: Information Sciences Institute, University of Southern California, 2004.
- [8] Shen Wei-Min, Chuong Cheng-Ming, Will P. Simulating Self-Organization for Multi-Robot Systems [D]. Los Angeles: Information Sciences Institute, University of Southern California, 2002.

(上接第 142 页)

流,从而把正在运行中的工作流迁移至新工作流,适应工作流程发生变化的实际问题。

参考文献:

- [1] van der A W M P, Jablonski S. Dealing with workflow change: identification of issues and solutions [J]. International Journal of Computer Systems, Science, and Engineering, 2000, 15(5): 267-276.
- [2] Sadiq S, Sadiq W, Orlowska M. Pockets of flexibility in workflow specifications [C] // Proceedings of the 20th International Conference on Conceptual Modeling. Yokohama, Japan: Springer, 2001: 513-526.
- [3] Casati F, Ceris, Pernicib, et al. Workflow evolution [C] // Proceedings of the 15th International Conference on Conceptual Modeling. Cottbus, Germany: Springer, 1996: 438-455.
- [4] Kradol Fer M, Geppert A. Dynamic workflow schema evolution based on workflow type versioning and workflow migration [C] // Proceedings of the 4th IFCIS International Conference on Cooperative Information Systems. Edinburgh, Scotland: IEEE Computer Society, 1999: 104-111.
- [5] 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.
- [6] 邓水光.动态工作流建模方法的研究与设计[J].计算机集成制造系统,2003,10(6):601-607.