

柔性工作流程动态建模机制的研究与设计

曹健辉¹, 李 峰²

(1. 上海交通大学 信息安全学院, 上海 200240;

2. 江苏省兴化市公安局网监大队, 江苏 兴化 225700)

摘 要:传统的工作流系统灵活性不强, 缺乏对动态性和自适应性需求的支持, 不足以应付复杂多变的业务流程。为了使工作流系统能够较好地适应变化, 对基于规则的柔性工作流的动态建模机制进行了深入的研究, 分析了现有的基于规则的工作流动态建模机制的不足, 提出了能够动态构建出包含循环逻辑的工作流模型的方法。这一方法能够高效地动态构建较为复杂的工作流模型。文中较详细地阐述了这一方法的具体步骤, 并结合实例展示了该方法的具体执行过程。

关键词:柔性工作流; 动态建模; 基于规则; 柔性活动

中图分类号: TP311.5

文献标识码: A

文章编号: 1673-629X(2009)10-0047-04

Research and Design on Dynamic Modeling Mechanism of Flexible Workflow

CAO Jian-hui¹, LI Feng²

(1. School of Information Security, Shanghai Jiaotong University, Shanghai 200240, China;

2. Net Supervise Battalion, Public Security Bureau, Xinghua City, Jiangsu, Xinghua 225700, China)

Abstract: Traditional workflow system is not flexible and not able to meet the need of dynamicity and adaptivity fully, so it is difficult to handle complicated and versatile transactions. To make the workflow system more flexible, conducts an in-depth research on the dynamic modeling mechanism of flexible workflow, analyses the drawback of present dynamic modeling mechanism and proposes a method which could be used to construct workflow model including loop dynamically. It possible to construct complicated workflow model efficiently. Lists detailed procedures of this method and uses examples to demonstrate the process of this method.

Key words: flexible workflow; dynamic modeling; rule-based; flexible activity

1 柔性工作流

1.1 柔性工作流定义

工作流柔性是指在不中断业务流程执行情况下, 随企业流程的变化可根据工作流活动和相关数据的实际情况动态改变执行路径, 具有自适应的能力^[1]。

1.2 柔性工作流特点

柔性工作流具有以下特点^[2]:

灵活性:可以灵活改变业务过程的模型。

集成性:对各种分散的应用都可以通过工作流管理系统进行集中管理和控制。

重用性:过程活动的实现和过程模型的重用。

可扩展性:应用开发和执行的规模可变化, 系统的结构可扩充。

2 基于规则的工作流动态建模机制

基于规则的柔性工作流模型的设计思想是, 在基本工作流体系结构的基础上引入规则库系统, 工作流在执行过程中根据规则库中提供的相应过程定义规则和数据来控制工作流的流向, 从而使业务过程能够自动、准确地完成^[3]。

2.1 柔性工作流模型

在柔性工作流模型中, 活动分为两种: 一是在工作流模型建模阶段能明确其完整定义的普通活动; 二是无法事先给出完整定义的柔性活动, 这种活动代表一个活动或者子流程^[4]。

本模型中, 一个柔性工作流是用一个四元组 $\langle ID, D, A, E \rangle$ 来表示的。其中, ID 为动态工作流的惟一标识符; D 为柔性工作流的基本信息, 如工作流的创建日期、描述信息等; A 为工作流流程中的所有活动构成的集合; E 表示该工作流的 ECA 规则集合^[5]。

工作流的每个活动可以表示成三元组 $\langle name,$

收稿日期: 2009-02-17; 修回日期: 2009-04-27

作者简介: 曹健辉(1984-), 男, 江西都昌人, 硕士研究生, 研究方向为工作流技术及应用; 李 峰, 研究方向为信息安全; 导师: 陈亚骏, 副教授, 研究方向为工作流技术研究、嵌入式系统开发。

type, SACS>。其中, name 是活动的名称; type 是活动的类型, 为普通活动或者柔性活动; 当活动类型为普通活动时, SACS 为空; 当活动类型为柔性活动时, SACS 表示在柔性活动具体化过程中必须遵循的规则, 它可以表示成二元组 $\langle AP, CC \rangle$, 其中 AP 为在柔性活动具体化过程中可选取的活动组成的集合; CC 为约束将从 AP 里选择来的活动组合成子流程的约束集。为了定义约束集, 定义如下操作符:

$a-b$: 活动 a 既能在活动 b 之前执行, 也能在活动 b 之后执行, 这两个活动之间可以有其他活动执行也可以没有。

$a \rightarrow b$: 活动 a 只能在活动 b 之前执行, 中间可以有其他活动执行也可以没有。

$a = > b$: 活动 a 只能在活动 b 之前执行, 中间不能有其他活动执行。

$a\Delta$: 活动 a 为循环逻辑的跳出节点。

$a \rightarrow b \rightarrow c\Delta \rightarrow d \rightarrow \dots m \rightarrow a$: 表示一个循环序列, 循环的进入节点为节点 a , 跳出节点为 c 。

2.2 基于规则的工作流动态建模方法

目前有很多的学者对基于规则的工作流动态建模方式进行了大量的研究, 给出了较好的动态建模方法。文献[4]中提出的方式步骤如下:

(1) 对于组合约束集里每一个约束, 如果这个约束里的活动不是全部被包含在活动集 A 中, 则停止这一过程, 否则继续。

(2) 判断当前约束的类型, 如果当前约束的类型是 $a-b$ 类型的, 则用一条无向直线连接约束的两个节点, 如果是 $a \rightarrow b$ 类型的, 就用一条有向线段从 a 节点指向 b 节点, 否则如果是 $a = > b$ 类型的, 也是用一条有向线段从 a 节点指向 b 节点。

(3) 对于以上步骤生成的图中的无向边, 对其方向进行枚举, 于是会产生出多个有向图, 如果原图中有 N 条无向边, 则会产生 2^N 个初始图。

(4) 对于所有的这些初始图, 对每个图中的所有节点进行分析。如果被分析的节点入度和出度均为零, 则用一条有向线段从起始节点连接到这个节点, 并用另一条有向线段从这个节点连接到终止节点, 否则如果该节点的入度为零, 则用一条有向线段从起始节点连接到这个节点, 如果该节点的出度为零, 则用一条有向线段从这个节点连接到终止节点。

(5) 对以上步骤生成的图中所有的节点都分析完后, 检测该图里是否包含环, 如果不包含则将这个图加入到备选集里, 这一过程直到所有初始图全部被分析完毕后结束。

(6) 对备选集里的所有图进行分析, 选出并行度最

大的作为最终构造好的工作流模型图。

这一方法的时间复杂度为 $O(N2^N)$, 但在实际应用中由于规则里像 $a-b$ 这样形式的规则不多, 故一般效率较高。但这个方法有一个很大的缺点, 这一方法无法动态根据规则构造出包含循环逻辑的工作流模型, 而在实际的工作流应用中循环逻辑的应用非常普遍, 因此这一缺点使得这一方法的适用性不强。

文中在对文献[4]中提出的动态建模方式的分析的基础上提出了可以构建包含循环逻辑的工作流模型动态构建方法。这一方法的具体步骤如下:

1) 判断规则中是否指明需要构造循环逻辑, 这可以通过检查规则中是否有 $a \rightarrow b \rightarrow c\Delta \rightarrow d \rightarrow \dots m \rightarrow a$ 这样形式的规则, 无则可以仍然按照文献[4]的方法构造工作流模型, 有则先构造循环逻辑, 无则按照前面提到的动态构造不包含循环逻辑的工作流模型的方法执行动态构造过程。这里构造循环逻辑的过程是: 对于 $a \rightarrow b \rightarrow c\Delta \rightarrow d \rightarrow \dots m \rightarrow a$ 规则中对应的 a 节点之前加上一个或聚合节点, c 节点之后加上一个或分支节点, 将节点 m 指向节点 a 的有向边改为由节点 m 指向或聚合节点, 节点 c 指向节点 d 的有向边改为由节点 c 指向或分支节点, 增加由节点或分支节点指向节点 d 的有向边和由或聚合节点指向节点 a 的有向边。最后再将该规则中其余对应的边全部加上去。这一过程的具体实现可以先把 $a \rightarrow b \rightarrow c\Delta \rightarrow d \rightarrow \dots m \rightarrow a$ 这样的循环规则展开成简单规则集再进行替换之后连接节点间有向边。

2) 将 CC 规则集里剩下规则中所有指向节点 a 的规则全部替换为指向或聚合节点的规则, 而所有节点 c 指向其他节点的规则全部替换为或分支节点指向这些其他节点的规则。

3) 按照规则集里的所有剩余规则连接节点间的有向边, 连接规则为: 如果当前规则的类型是 $x-y$ 类型的, 则用一条无向直线连接约束的两个节点, 如果是 $x \rightarrow y$ 类型的, 就用一条有向线段从 x 节点指向 y 节点, 否则如果是 $x = > y$ 类型的, 用一条有向线段从 x 节点指向 y 节点。这里为了便于分析只考虑 $x \rightarrow y$ 类型的规则, 其余规则的分析类似。这里的连接方法为:

(1) 如果规则为 $i \rightarrow j$ 且 i 和 j 均不被包含在任何循环里则直接用一条有向边由节点 i 指向 j 。

(2) 如果规则为 $i \rightarrow j$ 且 i 属于某循环但 j 不属于任何循环, 此时无法判断节点 i 执行后是选择执行还是并行执行各个分支, 这个时候需要用户的人工选择, 如果用户选择的是选择执行, 则需要增加一个或分支节点并将节点 i 指向原后继节点的有向边改为指向这个新增的或分支节点, 增加一条由该或分支节点指向节点 i 原后继节点的有向边和一条由该或分支节点指

向节点 j 的有向边。如果用户选择的是并行执行,则增加一条由节点 i 指向节点 j 的有向边。

(3) 如果规则为 $i \rightarrow j$ 且 j 属于某循环但 i 不属于任何循环,这种情况和规则为 $i \rightarrow j$ 且 i 属于某循环但 j 不属于任何循环的分析类似,在此不再赘述。

4) 对图中所有循环进行分析,如果循环内的某节点有并行输出分支指向不包含在循环里的节点,则对以这一节点的不被包含在循环里的后继节点为根的树形结构进行遍历,当第一次遍历到某一叶节点时,无法决定这一叶节点的后继节点具体是哪个节点,但这个后继节点必须在循环逻辑里,此时又需要用户的人工选择,用户做完选择后,增加一条由该叶节点指向用户指定节点的有向边,在之后的对整棵树的遍历过程中,每当遍历到叶节点时均增加一条由该叶节点指向用户指定节点的有向边。

5) 对图中所有节点进行分析,并按如下情况分别进行操作^[6]:

(1) 如果节点为或分支节点而且出度为一,则用一条有向线段从该节点指向终止节点。

(2) 如果节点为或聚合节点而且入度为一,则用一条有向线段从起始节点指向该节点。

(3) 如果被分析的节点入度和出度均为零,则用一条有向线段从起始节点连接到这个节点,并用另一条有向线段从这个节点连接到终止节点。

(4) 如果该节点的入度为零,则用一条有向线段从起始节点连接到这个节点。

(5) 如果该节点的出度为零,则用一条有向线段从这个节点连接到终止节点。

3 动态建模实例

考虑如下实例:

$A = \{a1, a2, a3, a4, a5, a6, a7\}$

$CC = \{cc1, cc2, cc3, cc4, cc5\}$

$cc1 = a1 \rightarrow a2, cc2 = a2 \rightarrow a4 \Delta \rightarrow a5 \rightarrow a2,$

$cc3 = a2 \rightarrow a3, cc4 = a3 \rightarrow a6, cc5 = a3 \rightarrow a7$

在动态构造工作流模型时,首先将循环逻辑先构造完毕,如图 1 所示。

接下来规则替换后按照剩余的规则连接其余节点,当遇到 $cc3$ 这个规则时,可以判断出 $a2$ 节点已经是某一循环逻辑里的节点,而 $a3$ 是循环逻辑外的节点,这个时候无法判断 $a2$ 与 $a4$ 间的分支和 $a2$ 与 $a3$ 间的分支是并行执行还是选择执行,因此需要用户来人工选择。

如果用户选择的是选择执行,则在步骤 3 中要在 $a2$ 之后加上一个或分支节点,并将 $a2$ 指向 $a4$ 的有向

边改为 $a2$ 节点指向该或分支节点的有向边,增加一条由该或分支节点指向 $a4$ 的有向边和一条由该或分支节点指向 $a3$ 的有向边,这一修改完成后如图 2 所示。这种情况下不需要执行步骤 4,按照步骤 5 将其它的边全部补齐得到图 3。

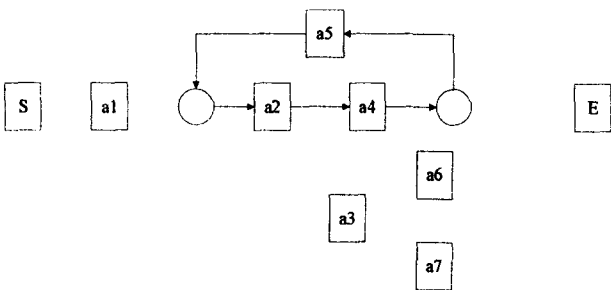


图 1 实例执行完步骤 1 后

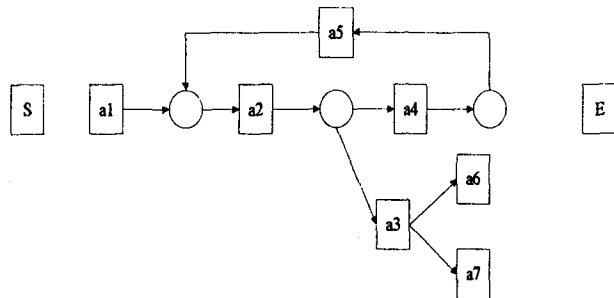


图 2 实例执行完步骤 3 后(选择执行)

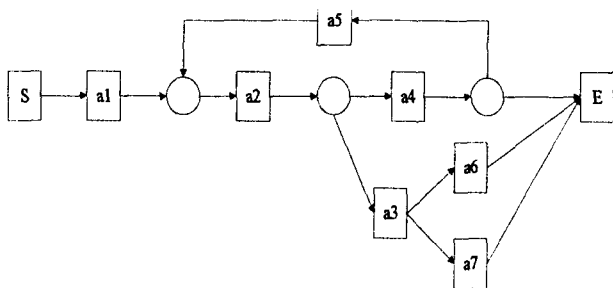


图 3 实例执行完步骤 5 后(选择执行)

而如果用户选择的是并行执行,则在步骤 3 中要在 $a2$ 节点之后加上一条指向 $a3$ 节点的有向边,完成步骤 3 后如图 4 所示。在步骤 4 里,因为节点 $a2$ 之后各个分支是并行执行的,需要对以 $a3$ 为根的树进行遍

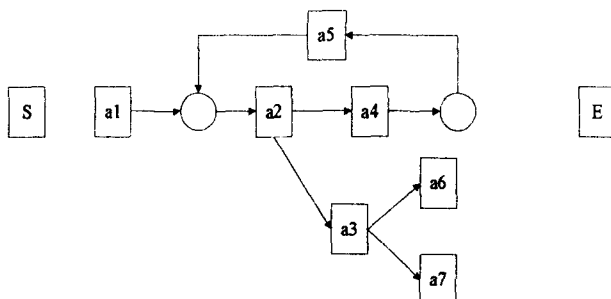


图 4 实例执行完步骤 3 后(并行执行)

历,当第一次遍历到树的叶节点 a_6 或 a_7 时,因为节点 a_6 和 a_7 执行后必须要回到 a_2 所在的循环逻辑里,这个时候无法判断应该回到循环逻辑里哪个节点处,需要用户人工选择,假设用户选择节点 a_6 和节点 a_7 执行完毕后均转移到节点 a_4 处,则此时需要加上两条有向边分别由节点 a_6 和节点 a_7 指向节点 a_4 ,此时如图 5 所示,按步骤 5 将所有剩余边全部补齐后如图 6 所示。

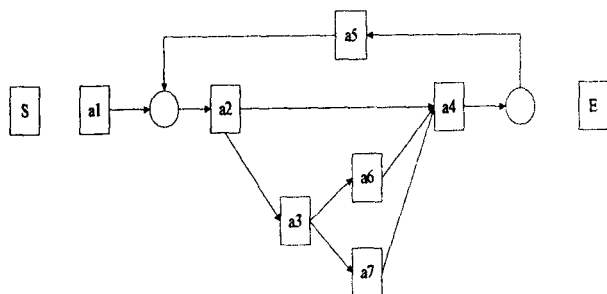


图 5 实例执行完步骤 4 后(并行执行)

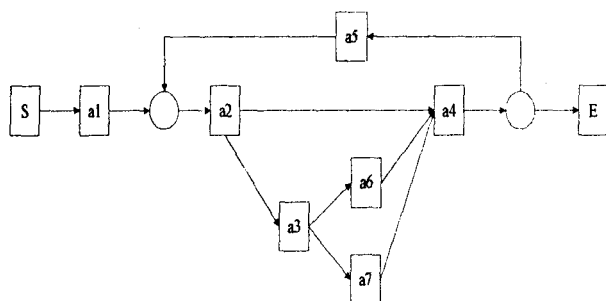


图 6 实例执行完步骤 5 后(并行执行)

(上接第 46 页)

ϵ 自进化是文中改进 GRASP 最大的特色,通过系统性地改变邻域结构来克服基本局域搜索算法易于陷入局部最小的不足,能有效地增强算法的勘探能力,提高算法获得全局最优解的可能性。这一点算例结果可以验证。

本算例中是采用固定的贪婪系数 α , Mockus 等人提出了有反应的 GRASP,其 α 值是自调节的,可以定期地根据近期得到的解的质量改变 α 值,获得更好的自适应效果。结合文中论述的方法可以作为后续研究的基础。

参考文献:

- [1] Feo T A, Resende M G C. A probabilistic heuristic for a computationally difficult set covering problem[J]. Operations Research Letters, 1989, 8(4): 67-71.
- [2] Feo T A, Bard J F. Flight scheduling and maintenance base planning[J]. MGMT Science, 1989, 35(12): 1415-1432.
- [3] Feo T A, Resende M G C, Smith S H. A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set[J].

4 结束语

这一动态建模过程较为复杂,时间复杂度为 $O(N^2 2^N)$,但在实际应用中由于规则里像 $a-b$ 这样形式的规则不多,故时间复杂度一般为 $O(N^2)$,效率还是比较高的。改进的动态建模方法能够很好地构造出包含循环逻辑的工作流模型,但这一方法还不适合复杂的多循环逻辑间转移的工作流模型,如模型中一个循环逻辑里有一节点指向另一个循环逻辑里的节点。

参考文献:

- [1] Kammer P J. Techniques for supporting dynamic and adaptive workflow[J]. Computer Supported Cooperative Work, 2000, 9(3,4): 269-272.
- [2] 尹建伟,陈刚,董金祥. 柔性工作流动态行为建模方法[J]. 计算机辅助设计与图形学学报, 2002, 14(6): 31-37.
- [3] van der Aalst W M P, Basten T. Inheritance of workflows: An approach to tackling problems related to change[J]. Theoretical Computer Science, 2001, 24(2): 125-203.
- [4] 邓水光,吴朝晖,余镇. 支持动态建模的工作流管理系统的研究与设计[J]. 计算机辅助设计与图形学学报, 2004, 16(5): 712-719.
- [5] 刘利,徐汀荣. 一种基于 BPEL4 WS 的柔性化工作流模型的研究[J]. 计算机技术与发展, 2006, 16(2): 187-190.
- [6] 陈张,赵卫东. 基于有向图的工作流模型的研究[J]. 计算机技术与发展, 2006, 16(4): 69-72.
- [7] Operation Research, 1994, 42(5): 860-878.
- [4] Resende M G C. Computing Approximate Solutions of the Maximum Covering Problem with GRASP[J]. Journal of Heuristics, 1998, 4(2): 161-177.
- [5] Marti R. Arc crossing minimization in grasps with GRASP[J]. IIE Transactions, 2001, 33(10): 913-919.
- [6] 金华征,程浩忠,夏夷,等. 贪婪随机自适应搜索法在电网规划中的应用[J]. 上海交通大学学报, 2006, 40(4): 563-567.
- [7] Prabhakaran G, Khan B S H, Rakesh L. Implementation of grasp in flow shop scheduling[J]. The International Journal of Advanced Manufacturing Technology, 2006, 30(11): 1126-1131.
- [8] Binato S, Hery W J, Loewenstern D M, et al. A GRASP for Job Shop Scheduling[R]. [s. l.]: AT&T Labs Research Technical Report, 2000.
- [9] Pitsoulis L S, Resende M G C. Greedy Randomized Adaptive Search Procedures[R]. [s. l.]: AT&T Labs Research Technical Report, 2001.
- [10] 金华征. 考虑市场环境的多目标输电网规划优化目标[D]. 上海: 上海交通大学, 2007: 84-94.