

文章编号: 1006-5911(2002)03-0202-05

workflow 管理系统实现技术研究

李伟平, 李莉, 薛劲松, 陈静杰

(中国科学院沈阳自动化研究所, 辽宁 沈阳 110016)

摘要: 利用有向图定义了一种 workflow 管理系统中的面向对象的工作流模型, 给出了该模型的执行算法和状态转换关系, 以及资源竞争时的调度策略, 设计了利用面向对象技术实现该模型的一种方法。利用该模型和相关技术, 给出了一种 workflow 管理系统结构, 并在软件中加以实现。

关键词: workflow 管理系统; workflow 模型; 过程定义; 面向对象; 有向图

中图分类号: TH166 **文献标识码:** A

从本质上讲, workflow 是使在多个参与者之间按照某种预定义的规则自动地传递文档、信息或任务的过程, 以实现某个预期的目标, 或促使此目标的实现^[1]。在 workflow 管理系统中, 过程定义负责将企业的实际经营过程和生产过程转化为计算机可处理的工作流模型。在对企业进行过程定义时, workflow 模型的优劣直接决定了 workflow 管理系统能否准确、合理、高效地反映企业的结构。目前, 已经出现了各种模型定义方法^[2-5], 它们各有特点。但对于模型在 workflow 管理系统中实施的一些关键技术问题, 如模型的状态转换关系、调度策略等研究还远远不够。

本文在总结前人工作的基础上, 定义了一种 workflow 管理系统的面向对象的工作流模型, 详细描述了该模型的执行算法、状态转换关系, 以及资源竞争时的过程调度策略, 给出了利用面向对象技术实现该模型的一种方法, 并利用上述模型, 设计和开发了 workflow 管理系统 Inworks。

1 workflow 模型

一个企业从设计、生产到销售、管理等各个阶段, 都可看作是一系列的过程。企业过程包括营销过程、产品设计过程、资金流过程、物流过程和计划流过程等^[6]。企业是这些过程组成的有机体。要

实现对这些过程的管理, 必须首先建立这些过程的模型, 即进行过程定义, 给出 workflow 模型。过程定义是 workflow 管理系统中最基本的组成部分, 也是 workflow 管理系统为企业用户服务的核心。不同的过程定义构成不同的 workflow 管理系统。过程定义的结果是给出企业的工作流模型。下面给出 workflow 模型的定义。

在本文中, 用有向图来表示 workflow 模型。有向图中的各个结点表示 workflow 过程中的各个活动, 有向边表示各个结点(活动)之间的控制关系。

定义 1 结点 N 是过程中的一个活动, 即一个工作步骤, $N = \langle n, t, d, u, m, \text{preA}, \text{postA}, \text{preC}, \text{postC}, T \rangle$, 是一个十元组。其中, n 表示结点名称; t 表示活动类型, 结点类型分为原子结点 a 和子过程结点 p ; d 是用到的数据集合, 即结点的参数集合, 包括活动执行时用到的输入、输出数据, 所用到的数据都保存在相应的数据库中, 数据库包括组织模型库和企业资源库, 活动之间通过数据库和电子文档两种方式传递信息; u 表示活动的执行者, 可以是人员、计算机或其他机器; m 表示手动执行或自动执行的执行模式; preA 表示与当前结点相连接的前驱边集合, 以及每条边的控制流程; 可能的控制流程有: 串联、与分叉、或分叉; postA 表示与当前结点相连接的后继边集合, 以及每条边的控制流程,

收稿日期: 2001-04-02; 修订日期: 2001-06-19。

基金项目: 国家自然科学基金重大资助项目(59990470; 69884005)。

作者简介: 李伟平(1973-), 男, 辽宁凌源人, 中国科学院沈阳自动化研究所博士研究生, 主要从事 workflow 管理系统、ERP 和企业建模等研究。

E-mail: lw p@ms.sia.ac.com。

可能的控制流程有: 串联、与合并、或合并、迭代。在此, 前驱边与后继边组合起来, 控制流程有五种情况: 串联、与分叉一与合并、与分叉一或合并、或分叉一或合并、迭代; $preC$ 表示当前结点的前置条件; $postC$ 表示当前结点的后置条件; T 表示结点执行所需的时间, 即活动的持续时间。

定义 2 原子结点 a 是一个不能再分的结点。

定义 3 子过程结点 p 是可以细化成另一个过程的结点。

定义 4 有向边 C 描述活动之间的数据流动关系和控制关系。 $C = \langle N1, N2 \rangle$ 表示从 $N1$ 到 $N2$ 的一条有向边。 $N1$ 到 $N2$ 分别为边 C 的前驱结点和后继结点。

定义 5 C^- 为结点 N 的前驱边集合: $C^- = \langle C | C.N2 = N \rangle$ 。

定义 6 C^+ 为结点 N 的后继边集合: $C^+ = \langle C | C.N1 = N \rangle$ 。

定义 7 有向图 G 表示一个过程, $G = \langle n, N, C \rangle$ 为一个三元组, n 是过程的名字; N 是所有活动的集合, 在此处称为结点, C 是全部有向边的集合, 它描述各活动之间的控制关系。

2 工作流实例的执行、调度与状态控制

通过过程定义, 给出企业工作流模型——企业过程的形式化描述。该模型只是一个静态模型, 还不能直接用于 workflow 管理系统的执行。通常, 将工作流的一次执行过程称为一个工作流过程实例, 将工作流过程实例中的一个活动称为活动实例。要想在 workflow 管理系统中运行工作流模型, 首先要建立该工作流模型的一个实例, 然后给出该实例运行之前所需要的全部数据, 最后才能运行该实例。也就是说, workflow 管理系统中运行的是模型的实例, 而不是模型本身。下面给出实例的定义。

定义 8 活动实例 Ni 是过程实例中的一个工作步骤, $Ni = \langle n, st, et, s \rangle$ 。其中, n 表示与此活动实例相对应的活动的名称。 st 表示 Ni 的开始时间。 et 表示 Ni 的结束时间。 s 表示 Ni 的状态, 它可能的状态有初始、执行、就绪、等待、挂起和完成。在创建了活动实例之后, 结点 N 的属性自动地赋给了 Ni 。

定义 9 过程实例 Pi 是工作流的一次执行过程, $Pi = \langle n, np, st, et, s, Pri \rangle$ 。其中, n 表示过程实例的名字; np 是相对应的工作流模型的名字; st 是过程实例的开始时间, et 是过程实例的结束时间,

s 是过程实例的状态, 它可能的状态有初始、运行、完成、终止; Pri 是过程实例的优先级, workflow 机利用它进行过程间的调度。根据以上定义, 我们给出工作流实例的执行算法。

2.1 工作流实例的执行算法

根据活动实例和过程实例的定义, 给出工作流过程实例的执行算法如下。此算法由两部分组成: 一是过程实例和活动实例的创建, 二是活动实例的执行。

算法 1 用户根据过程 G 创建过程实例 Pi 和活动实例 Ni

(1) 从 workflow 管理系统中选择用户指定的过程 G 。

(2) 根据用户选择的工作流模型 G 以及指定的名字, 建立一个过程实例 Pi 和 Pi 中的所有活动实例 Ni , 并进行初始化操作:

$Pi \cdot n \leftarrow$ 实例名, $Pi \cdot np \leftarrow G \cdot n$, $Pi \cdot st \leftarrow$ 系统当前时间, $Pi \cdot s \leftarrow$ 初始状态; $Ni \cdot n \leftarrow N \cdot n$, $Ni \cdot s \leftarrow$ 初始状态。

(3) 找到 Pi 中与开始结点相对应的活动实例 Ni , 并进行初始化操作:

$Ni \cdot s \leftarrow$ 等待状态, 获得与 Ni 相对应的活动 N 的参数集合 d 。

(4) $Pi \cdot s \leftarrow$ 执行状态; $Ni \cdot s \leftarrow$ 就绪状态。

(5) 提交并执行 Ni 。

算法 2 活动实例 Ni 的执行

(1) 判断结点类型, 若为原子结点 a , 则至第 2 步; 若为子过程 p , 则记录过程实例的状态, 进入子过程的开始结点, 至第 2 步。

(2) 判断前置条件 $preC$ 是否满足, 包括 $d(N$ 的参数集合) 全部获得, 以及若存在 $preA$, 则 $preA$ 的状态都处于完成态(根据控制规则, 若与合并时要求所有的前驱边都处于完成态, 若或合并时, 则要求某一个前驱边为完成态即可), 若 $preC$ 为 $True$, 则执行下一步, 否则等待至条件满足。

(3) 判断 m , 若为手动执行, 则通知活动的参与者执行相应的动作; 若为自动, 则自动执行要求的操作, 并使 $st \leftarrow$ 系统当前时间, $s \leftarrow$ 执行状态。

(4) 判断是否执行结束, 对于手动执行, 等待用户通知活动结束; 对于自动执行, 若执行时间 $\geq t$, 则认为结束, 如果在执行期间发生异常终止命令, 则整个过程执行结束, 且过程状态为终止状态。若执行结束, 则判断后置条件 $postC$ 满足与否, 即是否根据 d 修改了数据库信息。若 $postC$ 不满足, 则等待;

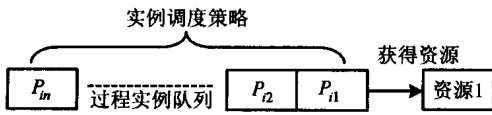


图3 资源竞争时的过程实例调度队列

本文根据过程实例的优先级进行调度。为使占用资源的时间最短,即 workflow 实例的平均流程时间最小,采用 SPT+ AT 规则来指定优先级。即:

(1) 首先运用 SPT (shortest processing time) 规则,过程实例的优先级为: $z_i = p_i$ 。其中, z_i 是过程实例 i 的优先级; p_i 是过程实例 i 占用资源的时间,且 $p_i \neq p_j$, 当 $i \neq j$ 。

(2) 当在两个 p_i 相等的情况下,采用 AT (arrival time) 规则,即:

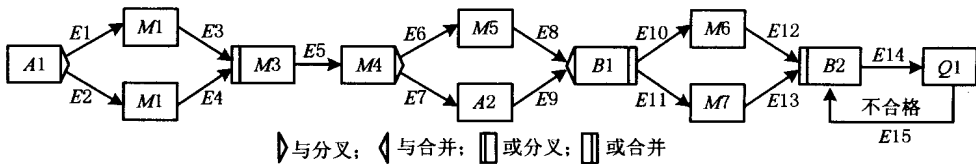
$$z_i = r_i, r_i \text{ 为实例 } i \text{ 的到达时间。}$$

在规则中, z_i 越小,实例的优先级越高,即 $z_i^* = \min z_i$ 。

2.4 workflow 实例的执行过程举例

根据以上定义,可给出一个具体的 workflow 模型例子,及 workflow 实例的执行过程。该实例是一个零件加工装配过程,如图 4 所示。

在过程开始时,由 A1 准备原材料, M1 和 M2 同时进行加工,因为在 M3 之前, E3 和 E4 是或合并关系,所以只要其中一个边完成, M3 就可以进行加工。实际上 M3 在 M1 或 M2 之后加工,对应两道不同的工序。M3 完成之后,交给 M4 继续加工。M4 完成之后,同时进行两个工作: M4 加工的半成品继续在 M5 上加工, A2 准备其他零件,在 M5 和



▷与分叉; ◁与合并; ◻或分叉; ◻或合并

图4 某零件加工过程

M—加工; A—准备原材料和外购件; B—装配; Q—质检

3 面向对象的工作流模型实现方法

根据企业 workflow 模型,采用面向对象的设计方法进行建模。从 workflow 模型定义可知,过程由结点和它们之间的联系构成,所以把结点和有向边分别用两个基类来描述: 结点类 CNode 和联系类 Cconnection,把过程用过程类 CProcess 描述。利用面向对象方法的继承性,可使不同的结点类型和联系类型从这两个基

A2 都完成以后,由 B1 进行装配,装配完毕,根据具体情况启动 M6 或 M7 继续加工。在 M6 或 M7 中的一个完成以后, B2 进行一次装配,最后由 Q1 进行质量检验,若不合格,则由 B2 重新装配。下面以图 4 中的零件加工过程为例,说明过程实例的执行和状态的转换过程。

在运行之初,首先产生过程实例及过程中的每一活动实例,并进行初始化。开始执行后,过程实例的状态变为执行态, workflow 机找到开始结点—A1,其类型为原子结点,判断前置条件 preC 是否满足,若满足则判断执行方式。结点 A1 为手动执行方式, $s \leftarrow$ 就绪状态。通知用户准备原材料,并且使 $st \leftarrow$ 系统当前时间, $s \leftarrow$ 执行状态,等待用户通知,若活动结束则检查后置条件,若满足则激活后继边 E1 和 E2,并使 $et \leftarrow$ 系统当前时间, $s \leftarrow$ 完成状态,执行结束。通过后继边 E1 和 E2 分别通知结点 M1 和 M2,并且使它们的状态变为等待状态。结点 M1 和 M2 判断前置条件,若满足则 $s \leftarrow$ 执行状态,并且等待用户通知,若活动结束则检查后置条件,若满足则通知相应的后继边 E3 或 E4,并使结点 M1 或 M2 的 $et \leftarrow$ 系统当前时间, $s \leftarrow$ 完成状态,执行结束。

此后,通过 E3 或 E4 分别激活后面的 M3。假设 E3 首先完成,则结点 M3 的状态变为等待状态。然后 M3 判断前置条件,如果满足则进入就绪状态,等待 workflow 机的调度执行。依此类推,直至 Q1 完成。在 Q1 完成之后,如果对加工的零件不满意,则通过 E15 申请迭代执行结点 B2,直到满意为止。

类派生,从而达到重用公共部分代码。

结点类 CNode 是所有结点的基类,根据结点的类型不同,可从 CNode 类继承而得到下面的类: 原子结点 CAtomNode 和子过程 CSubProcess; 而从原子结点又可继承得到开始结点 CStartNode、结束结点 CEndNode 和一般活动结点 CActionNode。一般活动结点又可分为手动结点 CManualNode 和自动结点 CAutoNode。各个类之间的关系如图 5 所示。

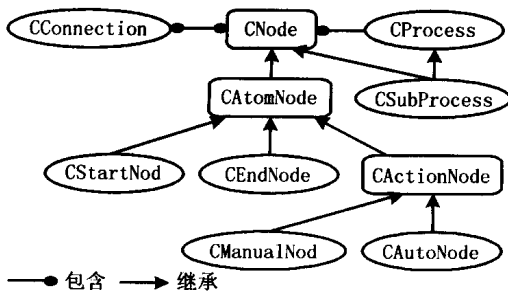


图5 工作流模型的类设计

4 工作流管理系统

参考 WfMC 给出的工作流管理系统体系结构, 以前述给出的工作流模型为基础, 我们设计了集成化工作流管理系统 Inworks。它是基于 Web 的、分布自治的工作流管理系统, 系统结构如图 6。

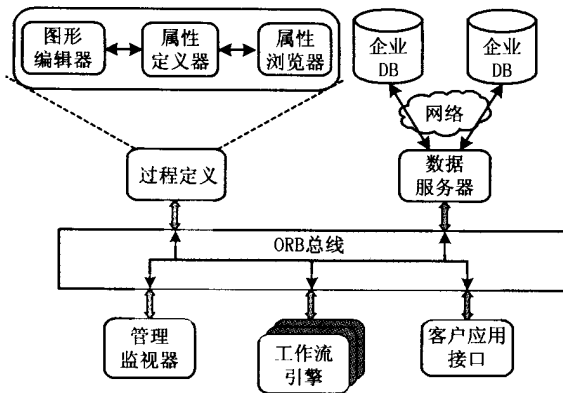


图6 Inworks系统结构图

由于 CORBA 软件总线具有良好的平台透明性和分布透明性, 以及分布对象操作能力, 可以解决企业的分布性和异构性。所以 Inworks 采用 CORBA 软件总线作为支撑平台, 实现了企业中各个自治系统接口的标准化, 解决了不同企业各个信息系统之间的异构性问题。Inworks 采用基于 Web 的结构, 由 ORB 软件负责具体的通信细节问题, 这样, 可以把精力放在工作流管理系统的功能上。

Inworks 工作流管理系统软件由五部分组成, 它们分别是过程定义器、工作流引擎、管理监视器、数据服务器和接口。

(1) **过程定义** 它是企业过程的计算机化的表示。过程定义器实现了对企业过程直观的、图形化的描述, 由图形编辑器、属性定义、属性浏览组成。

(2) **工作流引擎** 它是 Inworks 的核心部分, 负

责解释过程定义, 并进行过程实例化和过程执行控制、任务调度、日志维护等。此外, 它提供与工作流的参与者之间的接口, 必要时可激发其他 IT 工具和应用。

(3) **管理监视器** 对工作流的执行状态进行监视和控制。

(4) **数据服务器** 负责访问企业数据库。Inworks 中没有建立专用数据库, 而是通过分布式数据采集器获得企业的实际数据。企业在使用 Inworks 时, 可以充分利用企业现有资源, 直接建立工作流模型, 而不必重新建立数据库, 由此可减少工作量, 提高工作效率。

(5) **接口** Inworks 工作流管理系统提供了五个接口: ¹ 过程定义器与工作流引擎的接口; ^o 管理监视器与工作流引擎之间的接口; [»] 客户应用与工作流引擎之间的接口; ^¼ 数据服务器与工作流引擎之间的接口; ^½ 数据服务器与过程定义器之间的接口, 其中, 接口 ¹ 和 ^o 为内部接口, 接口 [»] ~ ^½ 为外部接口。

关于 Inworks 的详细情况可参见文献[7]。由于 Inworks 对过程的准确定义, 能容易地实现企业的过程集成, 从而对企业生产、经营过程进行管理。

参考文献:

- [1] Workflow M C. The workflow reference model [WfMC 1003] [R]. WfMC TC00 - 1003, 1994.
- [2] 史美林, 杨光信, 等. 一个基于 Web 的工作流管理系统[J]. 软件学报, 1999, 10(11): 1148- 1155.
- [3] 李红信, 薛劲松, 朱云龙. 集成化离散生产过程的工作流管理[J]. 计算机工程与应用, 2001, 37(4): 12- 15.
- [4] DICKERHOF M, DIDIC M M, MAMPEL U. CIMOSA —— background and case study [J]. Computer in Industry, 1999, (40): 197- 205.
- [5] LI Hongxin, XUE Jinsong, ZHU Yunlong. Multi-level description approach to flexible workflow model [A]. International Conference on Advanced Manufacturing System and Manufacturing Automation [C]. Guangzhou: International Conference on Advanced Manufacturing System and Manufacturing Automation, 2000. 661- 664.
- [6] 朱云龙, 薛劲松, 于海斌, 等. 面向对象的企业过程建模及实现方法研究[J]. 信息与控制, 2001, (5): 390- 395.
- [7] 李伟平, 薛劲松, 朱云龙. 敏捷制造环境下集成化工作流管理系统的设计与实现[J]. 信息与控制, 2001, (5): 401- 406.

CIMS 环境下机组排产作业计划的在线生成问题, 实现了生产线的生产路径优化控制, 提高了生产质量和设备利用率。到目前为止, 该软件已在现场稳定运行近一年, 机组作业计划几乎完全按照该软件生成的结果进行, 任务调整率不到 0.5%。该软件已成为鞍钢冷轧厂生产运行的关键组成部分。

5 结束语

本文通过鞍钢新轧钢集团股份有限公司冷轧薄板厂 CIMS 工程的实例, 介绍了详细设计阶段以冷轧生产线的软件架构模型进行生产线的软件功能设

计, 以及机组作业计划中的任务分配方法。该方法简单实用、效率高、效果好。实践证明, 该方法起到了事半功倍的作用, 是 CIMS 工程详细设计阶段和软件开发的有效方法。

参考文献:

- [1] ASSAF I, CHEN M, KATZBERG J. Steel production schedule generation[J]. International Journal, 1997, 35(2): 467- 477.
- [2] ARIZONO N, SAEKI T, et al. Expert system for scheduling rolling sequence in plate mill[J]. KOBELCO Technology Review, 1991, (10): 47- 49.

Software Framework and Tasks Distribution for Unit- Operation Planning

PENG Wei, SHI Hai-bo

(Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110015, China)

Abstract: The software function design and tasks distribution method for unit operation planning under CIMS environment are introduced, according to the real situation of cold rolling Mill of Angang New Steel Stock Ltd. With colored Petri Net, the software framework model of cold rolling product line is constructed, which completely describes the production flow of cold rolling product line and corresponded software's functions. In the unit operation planning's function software, load balancing and optimal matching between unit and production tasks are set as criteria. And the assigning rule of the least load distribution is adopted, to solve the problem on unit operation plan on line generation of the cold rolling product line under CIMS environment.

Key words: petri net; CIMS; cold rolling product line; software framework model; task distribution

Received 6 Apr. 2001; Accepted 26 Jun. 2001.

(上接第 206 页)

Workflow Management System Implementation Technology

LI Wei-ping, LI Li, XUE Jin-song, CHEN Jing-jie

(Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)

Abstract: We defined an object oriented workflow model using directed graph, gave the execution algorithm, status transition method and scheduling strategy existing resource conflict, and designed an implementation method using object-oriented technology for this workflow model. Using this workflow model and related technologies, we developed a workflow management system and the software.

Key words: workflow management system; workflow model; process definition; object-oriented; directed graph

Received 02 Apr. 2001; Accepted 19 Jun. 2001.

Foundation item: Project supported by the National Natural Science Foundation, China(Grant No. 59990470, 69884005).