

支持成批服务的工作流模型及事件驱动下的调度研究^{*})

刘建勋

(湖南科技大学计算机学院 湘潭411201)

摘要 在实际应用当中,成批服务一般是作为生产或者是商务流程中的一个或者数个环节而存在。工作流管理作为支持商务流程处理的有力工具,受到了学术界与产业界的广泛关注并有深入的研究。但是很遗憾,现有的工作流管理没有考虑到该问题,因此也未提供相应的支持机制。本文对工作流管理中引入成批服务问题进行了深入的研究,提出了一个可支持成批服务的工作流模型,并建立了一个基于事件驱动机制的调度解决方案。

关键词 工作流,工作流管理系统,工作流模型,成批服务,调度,事件驱动

Study on Workflow Model Supporting Batching Processing Services and its' Event-Driven Scheduling Mechanism

LIU Jian-Xun

(School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract In practical application, batch service commonly exists as one or several links of production or business process. Workflow management serving as a powerful tool, which supports processing of business process, has gained extensive attention from industry and research world, and has been studied deeply. But it is a pity that the workflow management at present hasn't taken the problem into consideration, let alone provide the relevant supporting mechanism. This paper has done some in-depth research to introduce batch service to workflow management, and proposed a workflow model which supports batch service. As well as provided a scheduling solution based on event-driven mechanism.

Keywords Workflow, Workflow model, Batch service, Scheduling event-driven

1 引言

随机服务系统是人们在日常生活中经常会遇到的一类服务系统,例如去快餐店就餐以及 Internet 上的各种服务等。各种随机服务系统具有下列共同组成部分:1)输入过程,它描述了各种类型的“顾客”按怎样的规律到来;2)排队规则,指到来的顾客按怎样的规定次序随机接收服务;3)服务机构,指同一时刻有多少服务设备可以接纳顾客,每一顾客服务多少时间。随机服务系统研究的目的是如何合理地设计与控制随机服务系统,使得它既能满足顾客的需求,又能使机构的花费最为经济^[1]。随机服务系统有多种类型,例如最简单的随机服务系统、M/G/1 系统以及特殊的随机服务系统等等。在特殊的随机服务系统中,成批服务的系统有着较广泛的应用背景,因为成批服务可以节约系统资源,提高系统处理效率^[2]。但是实际应用中,成批处理的系统一般不是独立存在,而是表现为生产和商务流程的一个或者数个环节,例如,生产加工流程中的喷漆、快餐店服

务流程中炒菜、订单驱动的商务流程中的配送等。工作流是支持商务流程处理的有力工具^[3~6],然而目前的工作流研究没有考虑到该问题,未提供这一方面的支持,因此扩展现有工作流模型及其执行机制以支持成批服务是很有必要,也是很有意义的。

本文在工作流管理中引入了成批服务这一思想,建立了可支持成批服务的工作流模型,并提供了一个基于事件驱动机制的调度解决方案。

2 成批服务及其优化调度问题分析

徐光辉^[1]将成批服务的系统描述为:假定参数为 λ 的最简单流到达单服务台的等待系统,空闲的服务台当且仅当等待的顾客数不小于 r ($r \geq 1$ 为一常数)时才开始服务,每次同时服务 r 人(同时开始服务,同时结束服务),服务时间与到达间隔独立。

然而实际应用系统比这要复杂,例如:1)到达的服务可能会有不同的类型,只有相同或相近的类型才可成批处理;2)服务会有到期时间(Deadline),在这个竞争日益激烈的社会中,对系统资源的节省不

^{*})本研究得到湖南省自然科学基金(项目号:03JJY6024)和湖南省制造业信息化项目(项目号:Hnmiie-A-051)资助。刘建勋 博士,副教授,副院长。主要研究方向为:电子商务,工作流管理系统,分布式计算环境等。

能以过分牺牲顾客的利益来实现,否则会失去顾客;
3)服务时间可根据服务数量基本估计或统计出来;
4)成批的服务不是独立存在的,而只是生产与商务领域中的一个环节。图1所示的某一经营盖浇饭的快餐店就是一个典型的该类系统。其盖浇饭业务处理

流程包括受理、收费、炒菜、配饭、派送流程,对于每一个客户,其就相当于一个订单。为了节约时间和资源,在炒菜时需要将不同客户的订单(同类型盖浇饭)组合在一起成批服务(炒菜)。

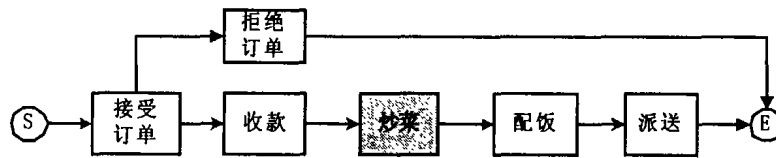


图1 盖浇饭处理流程

根据上述分析,本文将实际应用中具有上述特征的成批服务系统称为分组成批服务系统或称分组成批处理系统。对于单服务台的分批服务系统可由如下参数刻画:1)参数为 λ 的最简单服务到达流;2)服务有 m 种类型(其服务类型的到达服从某种概率分布);3)服务有到期完成时间 t_{DL} 要求;4)一个长度为 B_1 的缓冲区,用于存放等待的服务;5)一个一次最多可处理 C_p 个服务的服务机构(处理机);6)遵循参数为 μ 的负指数分布的服务时间。

对这种系统,评价的性能指标多种多样:例如处理机吞吐率 η_p 、平均等待时间 T_{avg} 以及服务到期未满足率 P_{us} (完成时间超过 t_{DL} 的服务占总服务数的比例)。然而这些优化目标之间是有矛盾的,例如 η_p 的提高会引发 T_{avg} 与 P_{us} 的增加。因此在实际应用中优化的目标为:在满足 T_{avg} 与 P_{us} 的条件下,使 η_p 最高。换句话说就是在不影响交货期的情况下,尽可能地提高出现成组服务的概率。

因此缓冲区 W_{Que} 的设置是十分必要的,只有保证在缓冲区中有足够长的服务等待队列,设队长为 Q ,才能使分批得以发生(因为只有相同类型的服务才可成批处理),但是过长的等待队列,会延长平均等待时间。因此如何设置等待队列,以及设置在什么时候处理机应该等待,而什么时候处理机必须恢复处理即很重要的问题。为此在优化调度时要引入缓冲区长度、调度报警下界、处理机等待时间等概念来描述分析问题。但对如何具体的设置这些参数,这不是本文研究的重点,因此在此不予多介绍。

•缓冲区长度 B_1 ,一般受到物理条件的限制,因此本文不作分析。

•调度报警下界 B_{lower} :利用 B_{lower} 作为影响处理机等待的因子,即 $Q \leq B_{lower}$ 时,处理机要等待一个随机时间 Δt_1 。设分组依服务(产品)类型进行,则 B_{lower} 的设置与 m 有关。假设不同类型的产品在到达时服从均匀分布,则必须使 $B_{lower} > m$,才可保证在缓冲区中有相同类型的服务(产品)存在,所以 m 应是 B_{lower} 的取值的下界。在实际应用时, B_{lower} 的值可通过仿真或者是根据经验得来。

•处理机随机等待时间 Δt_1 与 Q 以及等待服务请求的性质(如某个订单的交货期)有关,具体可根据实际应用的经验取值。

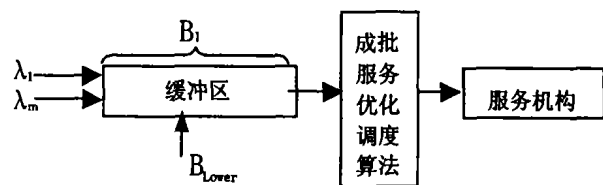


图2 单服务机构成批服务调度模型

根据上述分析,成批服务优化调度问题可抽象成图2所示的模型,图中用 $\lambda_1, \dots, \lambda_m$ 表示输入流是由 m 种不同服务类型的到达流复合而成。因此实际优化调度算法要考虑两个方面问题:1)如何对缓冲区中服务分组并选择一个分组提交给服务机构执行的分组优选算法,以下记为GSA(Grouping and Selection Algorithm);2)启动GSA的调度算法,以下记为GSA-SA(GSA Schedule Algorithm)。对于GSA优化算法,它与过程本身没有关系,也就是说,它与流程控制没有直接的联系,可以认为是一个具体的执行者行为,因此本文将不研究。而对于GSA-SA算法,由于其与流程控制有关(决定何时启动GSA算法),因此其是本文研究的重点之一。

3 支持成批服务的工作流模型

3.1 成批服务在工作流中存在的模式

根据前面的分析,成批的服务不是独立存在的,而只是存在于生产与商务领域中流程的一个或者几个环节,这些环节在工作流中表现为一个或几个活动,例如图1。因此这种引入成批服务的工作流类型可表现为图3所示的3种形式。

在图中设工作流类型 W 中有3个活动A、B、C,用虚线所框住的活动表示可成批服务,在此称其为成批服务区(BPA:Batching Processing Area),活动上方的文字表示该活动的执行者(即为该活动提供服务的实体),例如活动B的执行者即是 Agt_b ,并假设系统中存在工作流类型 W 的两个实例 w_1, w_2 。

则在图3a中,只有活动B需要考虑成批服务,图3c中尽管B、C均要考虑成批服务,但是它们仍然是各个自为政,即每个活动的成批服务是相互独立的,彼此不存在影响,例如在活动B处,执行服务将 w_1 , w_2 的活动实例 $w_1.b$, $w_2.b$ 整合成一个活动实例 b' ,提交给 Agt_b 执行,但是在 Agt_b 执行完 b' 后,在 Agt_c 的输入队列(工作列表)中,仍然是两个独立的 $w_1.b$, $w_2.b$.在 Agt_c 处又要重新考虑分组,且可能使用不同的分组整合策略,这样 $w_1.b$ 与 $w_2.b$ 可能会整合到不同的分组中.图3b中的情况则与图3c不同,W中B、C将采用同一种分组方式,表示只在活动B处整合一次,然后在整个BPA中均将按照整合后的方式处理,一直到离开BPA后,才恢复原来的状态.即有 $w_1.b, w_2.b \in b'$ 与 $w_1.c, w_2.c \in c'$,它们分别表示 w_1, w_2 在B和C处均是以相同的方式整合在一起执行.它对于这种方式,在分组整合时要考虑整个BPA内为各活动提供服务的服务机构的服务能力与特性,因此分组优化处理调度算法将会比较复杂.

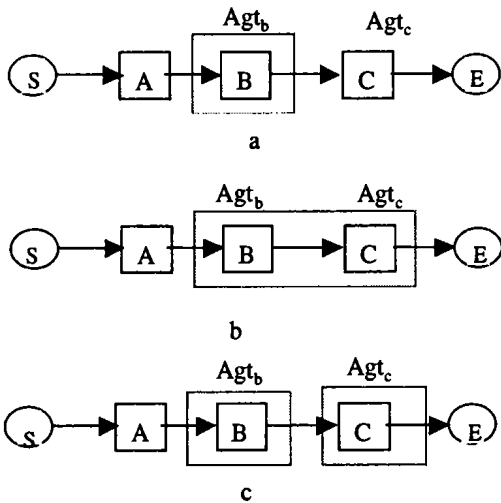


图3 工作中成批服务的不同方式

同时分组整合时还要考虑BPA内服务机构(活动的执行者 Agent)的数量问题,即多个参与者均有权执行某个活动.根据这些分析可总结出, workflow成批服务模型可表示为 $X/Y/Z$ 模型,其中 X 表示BPA的数量, Y 表示该BPA中活动的数量, Z 表示为BPA中活动提供服务的服务机构的最大数量.共有8种不同形式: $1/1/1$ 、 $1/1/N$ 、 $1/M/1$ 、 $1/M/N$ 、 $K/1/1$ 、 $K/1/N$ 、 $K/M/1$ 、 $K/M/N$.例如:

- $1/1/N$ 表示只有一个BPA,其中只有一个活动,但在该区中有2至 N 服务机构(执行者);
- $1/M/1$ 表示只有一个BPA,其中只有 $M(M > 1)$ 个活动,但在该区中每个活动的服务机构(执行者)为1;
- $K/1/1$ 表示有 $K(K > 1)$ 个BPA,每个BPA

中只有一个活动,且各BPA中活动的服务机构数量(执行者)为1;

- $K/M/N$ 表示有 $K(K > 1)$ 个BPA,每个BPA中可有 $M(M > 1)$ 个活动,且各BPA中活动的服务机构数量(执行者)为 $N(N > 1)$.

3.2 支持成批服务的工作流模型

成批服务对工作流模型与工作流管理系统带来了新的要求,因为分组批处理是动态的,并且要将活动实例整合成新的实例提交给参与者执行,因此必须要考虑整合的方法问题.为在工作流中支持成批服务,本文根据前面3.1节介绍的BPA概念,在工作流模型中引入一新的活动类型—成组处理活动类型,记为BPAT(BPA Type),并引入成组处理子过程的概念,对于每个BPAT必有一个相应的成组处理子过程与其对应.BPAT描述了活动实例的优化分组与调度的方法、策略,其定义如下:

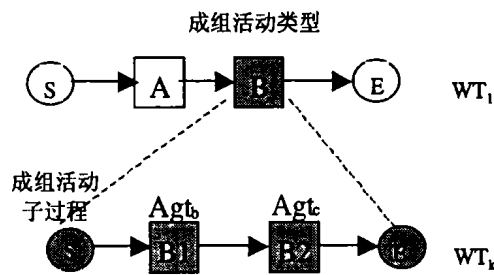


图4 成组活动类型与成组子过程的定义及其关系

定义1 成组处理活动类型BPAT为五元组, $BPAT = \langle GSA-SA, GSA, B_{Lower}, B_1, GC \rangle$,其中 $GSA-SA$ 、 GSA 就是前面介绍的 $GSA-SA$ 与 GSA 算法,对于 $GSA-SA$ 在后面还有详细的分析,此处不赘述; B_{Lower}, B_1 为与缓冲区有关的参数,其已在前面介绍; GC (Grouping Characteristic)为成批处理中的分组归类的特征.

通过增加一个成组活动类型,并建立一个相应的成组处理过程与其关联, workflow模型可以支持成批服务的定义.例如对于图3b的工作流类型,其新定义可见图4所示.在工作流类型定义时只需要在成组活动类型中设置 GSA 算法、调度 GSA 算法 $GSA-SA$ 、缓冲区的相关参数以及分组归类特征即可,其余的将由 workflow执行系统去完成.

4 支持成批服务的工作流执行系统研究

尽管在引入成批服务后, workflow模型这一级没有太大的变动(这主要是因为成批服务是动态的),但是在 workflow执行系统的实现上必须做较大的扩充.在 workflow执行系统的设计与实现中,很多均是使用基于事件驱动方式^[3,6,7].为此,对成批服务的工作流执行也应基于事件驱动的执行方式.下面将给予详细的分析.

4.1 执行时刻系统的新需求

归纳起来,引入成组处理后对 workflow 执行机制带来的变化主要包括如下方面:

1. 经过成组处理后,系统中具有相同特征的活动实例将会由一个新创建的活动实例替代。也就是说在成组处理子过程内部,参与者所看到的活动实例与进入和退出成组处理后的活动实例是不同的,如图5刻画了这种不同。设系统中活动实例不断产生,且 b_1, b_3, b_4 具有相同特征,则进入 BPA 后它们将被组合为一新的活动实例 b'_1 , 如果成组处理区包含多个活动类型,则它们依同样的规则组合,实际上相当于在系统中生成一个新的 workflow 实例,即为成组子 workflow 类型的实例。而在退出成批服务区后又将要拆分,不过此时,活动实例的到达顺序已改变,例如 b_2 已位于 b_4 之后;

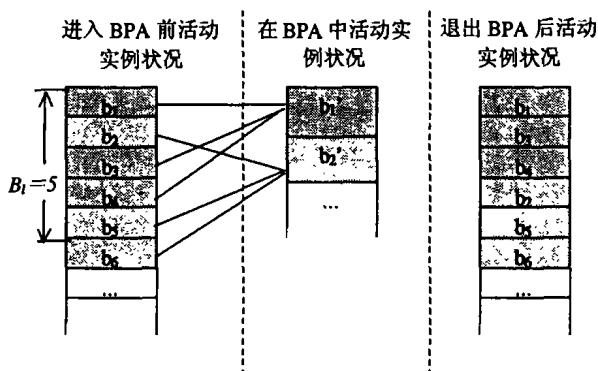


图5 BPA 内外活动实例的变化

2. 必须有专门的机制来负责活动实例的组合、拆分以及成组子 workflow 类型的实例化;

3. 引入成组处理后,活动的状态管理要作改变,因为将属于不同 workflow 实例的活动组合到了一起,并临时生成了一个新的内部的工作流实例,原有的 workflow 实例被暂停。因此必须建立内部临时的工作流实例中活动实例的状态与原来的活动实例之间的状态映射关系。

4.2 事件驱动的成批服务调度的设计与实现

根据前面分析,调度算法在不同的成批服务模式其复杂程度是不同的。对于 K/M/N 模型,优化调度本身是一非常复杂的研究问题,其研究不是本文的关注所在。本文以分析单服务机构的优化调度算法在 workflow 执行服务中实现问题,来说明如何通过扩展现有 workflow 执行服务以支持成批服务的实施。

4.2.1 单服务机构的成批服务 GSA-SA 算法研究

根据前面分析,GSA-SA 算法必需考虑来自服务机构与缓冲队列的状态信息。因此我们首先需要分析服务机构(workflow 执行者)与缓冲队列的状态变化图,分别见图6,7所示。从图中看出,状态的变化是由事件触发的,这是一个典型的事件驱动的有限

状态机模型。

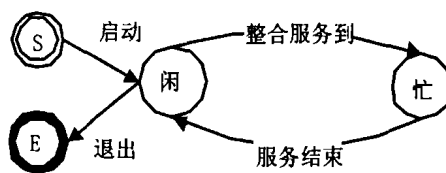


图6 服务机构状态转换图

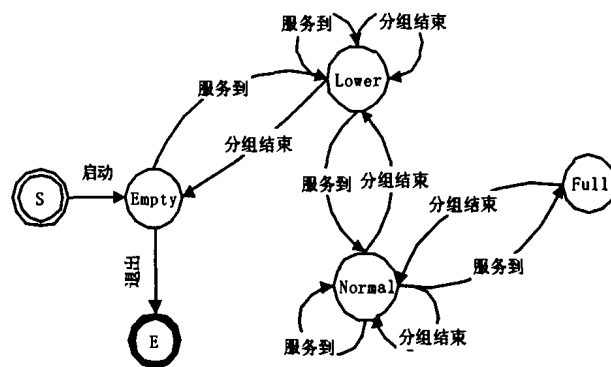


图7 缓冲区状态转换图

图6描述了服务机构的状态变化图,对于服务机构而言,其只有两种状态:“忙”,“闲”。“忙”表示服务机构正在提供服务,而“闲”则表示服务结束,服务机构在等待新服务的到来。图7描述了缓冲区的状态变化图,其有如下状态: Empty, Lower, Normal, Full, 分别表示缓冲区空、低于警戒下限、正常、满。Lower 与 B_{Lower} 有关,见第2节的说明。其状态转换为:在启动时,缓冲区为 Empty, 当有服务到时,则转入 B_{Lower} , 如果此时一直有服务到达,但等待队长 $Q \leq B_{Lower}$, 则仍是处于 Lower 状态,如果 $B_{Lower} < Q < B_1$, 则转入 Normal 状态,一直到 $Q = B_1$, 则进入 Full 状态。而在一个分组选优算法结束后会启动服务机构去处理分组 G_r , 用 $G_r \cdot Length$ 表示该分组中的服务个数, 则其相当于从缓冲队列中 W_{Que} 中删除 $G_r \cdot Length$ 个服务, 即 $Q = Q - G_r \cdot Length$, 此时有可能改变缓冲区的状态。它为服务到达的逆操作,因此在此处就不详述。

根据服务机构与缓冲区状态的变化,可以设计出基于事件驱动的 GSA-SA 算法。见图8所示的有限状态机变化图。算法启动后进入“空等1”状态,此时表示缓冲区为空($Q=0$), 如果此时有系统要求的退出事件(Event_Exit), 则退出该 GSA-SA; 如有 Lower 事件到(表示有服务到达,缓冲区非空), 则转入“空等2”状态,此时将设置一个定时器,在时间到后,产生等待时间到事件(End_Waiting-1)。算法一直维持在该状态,直到有 Normal 或者 End_Waiting-1 事件到,则进入“分组调度”状态,表示正启动

(下转第250页)

任务节点的属性、增加新的执行路径等,使得本文提出的 workflow 模型能够有效地支持某种“随机应变”的机制。

3.4 移动 Agent 的具体实现

workflow 应用的特点之一是支持企业信息流动。对此,移动 Agent 技术能够提供很好的支持,本文提出的 workflow 模型正是借助移动 Agent 的移动性和自治性,将执行每个 workflow 活动所需要的行为和 data 封装到移动 Agent 中,然后将它通过网络派遣出去。在该系统的具体实现中,采用基于 Java 的移动 Agent 系统 Aglet 来实现有关移动 Agent 的移动性和安全性(利用 com.ibm.aglets.Security 包)、系统设计样式(利用 com.ibm.aglets.pattern 包)、互操作性(利用 com.ibm.maf.MAFAgentSystem 包)等要求,并给出框架代码:

```
public abstract class Master extends Aglet{
    public void getResult();//得到移动 Agent 返回的信息
}
public abstract class Slave extends Aglet{
    public void run(){
        initializeJob();//初始化参数
        dispatch(destination);//分派到目标系统
        doJob();//在目标系统中执行任务
        dispose();//返回结果,然后清除
    }
}
下面是具体的调度 Agent (DispatchMaster 类)和具体的任务移动 Agent (ExecuteTask 类)代码:
public final class DispatchMaster extends Master{
    public getResult(Object result){
        //重载调度 Agent 中的 getResult 函数,把得到的结果存入相应的数据库中
    }
}
public final class ExecuteTask extends Slave{
    SimpleItinerary itinerary=null;
    protected void initializeJob()throws Exception{
```

```
//具体的初始化工作,如封装任务配置信息
}
protected void dispatch(String des){
    try{
        itinerary=new SimpleItinerary(this);
        itinerary.go(des,"doJob");//到达目标系统,执行下面的 doJob 操作
    }catch(Exception ex)
    {ex.printStackTrace();}
}
public void doJob()throws Exception{
    //在目标系统中执行具体任务
}
```

结束语 随着企业应用规模和功能的不断增大和调整,基于 workflow 技术的应用系统必将是将来软件技术的发展方向之一。该文通过分析当前 workflow 系统的一些不足之处,提出了一种基于 Agent 的自适应性 workflow 管理系统,研究了其中的一些关键算法和实现机制。论文未来的研究工作是与知识工程等领域的研究结合起来,进一步完善基于 Agent 的自适应性 workflow 管理系统。另外,增加 workflow 的自学习功能也是未来的研究方向之一。

参考文献

- 1 罗海滨,范玉顺,吴澄. workflow 技术综述[J]. 软件学报,2000,(11)
- 2 Workflow Management Coalition. The Workflow reference model [R], WFMX TC00-1003,1994
- 3 Chin Dickson K W, LI Qing, Karlapalem Kamalakar. A meta modeling approach to workflow management systems supporting exception handling [J]. Information Systems,1999,24(2):159~184
- 4 Arpinar S,Dogac A,Tatbul N. An Open Electronic Marketplace Through Agent-based Workflow: MOPPET. Intl. Journal on Digital Libraries,1998
- 5 Koksals P,Cingil I,Dogac A. A Component-based Workflow System with Dynamic Modifications. In:Proc. of the Next Generation Information Technologies and Systems(NGITS99),Istael,1999

(上接第245页)

GSA 算法,对 W_{que} 中的服务分组并选择一个分组提交给服务机构处理。分组调度结束后,服务机构忙,因此 GSA-SA 进入“忙等”状态,一直到服务机构完成服务并产生服务机构“闲”事件,此时转入“空等3”状态,等待与缓冲区有关的事件发生。根据缓冲区产生的不同事件 Lower、Empty、Normal 而分别转入“空等1”、“空等2”、“分组调度”状态。

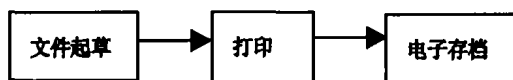


图8 分组调度算法状态转换图

结束语 成批服务的系统是一类特殊的随机服务系统,它有较广泛的应用背景。但是在实际当中,成批服务一般是作为生产或者是商务流程中的一个或者数个环节而存在。workflow 管理作为支持商务流程处理的有力工具,受到了学术界与产业界的广泛关注,但是现有的 workflow 管理没有考虑到该问题,因

此也未提供相应的支持机制。本文总结了 workflow 中出现成批服务的8种不同的 X/Y/Z 形式,建立可支持成批服务的 workflow 模型。分析了支持成组处理的 workflow 执行服务系统设计与实现会带来的新问题,利用状态图与事件的方法分析了 K/1/1 模型的优化调度问题。

参考文献

- 1 徐光辉. 随机服务系统. 北京:科学出版社,1980
- 2 王秀丽,吴锡华. 一种求解两机成组作业流水线车间优化调度问题的遗传算法. 系统仿真学报,2001,13(8):88~90
- 3 胡锦涛,张申生,余新颖. 基于 ECA 规则和活动分解的 workflow 模型. 软件学报,2002,13(3):761~767
- 4 刘建勋,胡涛,张申生. 基于 workflow 与 XML 的敏捷供应链管理系统集成框架研究. 计算机集成制造系统-CIMS,2001,7(6):6~35
- 5 罗海滨,范玉顺,吴澄. workflow 技术综述. 软件学报,2000,11(7):899~907
- 6 Goh A, Koh Y K, Domazet D S. ECA rule-based support for workflows. Artificial Intelligence in Engineering,2001,15(1):37~46
- 7 Liu Jianxun, Yao Yingxiong, Tang Xinhui. Pre-dispatching of tasks in workflow: the concept, model and implementation. Chinese Journal of Electronics,2004,13(2)