# 基于有色 Petri 网的工作流阶段性调度*)

肖志娇　常会友

(中山大学信息科学与技术学院计算机科学系　广州 510275)

**摘　要**　工作流的合理、有效调度有利于改善整个工作流系统的性能,从而提高业务流程的执行效率。静态调度有利于在静态环境下达到全局调度的最优,但不能有效地处理工作流的动态不确定性。而动态调度在考虑工作流的动态不确定性的同时,优化每个任务的调度方案,但很难达到所有任务的全局最优。在总结静态调度和动态调度两种方法各自的优缺点的基础上,本文提出了一种基于有色 Petri 网的工作流阶段性调度方法。该方法能够妥善地处理工作流的动态性和不确定性,并在静态全局最优和动态单个最优之间达到较好的均衡。仿真实验说明了该方法的有效性和优越性。

**关键词**　工作流,调度,有色 Petri 网

## A Method of Workflow Scheduling Based on Colored Petri Nets

XIAO Zhi-Jiao　CHANG Hui-You

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275)

**Abstract**　An effective method of workflow scheduling can improve the performance of the whole workflow system. Static methods of workflow scheduling can reach the global optimal results under static circumstance. But they cannot deal with uncertainties and dynamic circumstances. Dynamic scheduling methods can optimize workflow scheduling while taking all the uncertainties and the dynamic circumstances into account. The results achieved by dynamic methods are usually optimal for single task, but not optimal as a whole. Based on the analysis of static scheduling methods and dynamic ones, a new workflow scheduling method is proposed. The uncertainties and the dynamic circumstances can be well dealt with. And a satisfactory balance between static global optimization and dynamic local optimization can be achieved. Experimental results show the feasibility and the priority of this method.

**Keywords**　Workflow, Scheduling, Colored Petri nets

## 1　引言

工作流是一类能够完全或者部分地由计算机自动执行的经营过程,它根据一系列过程规则使得文档、信息或任务能够在不同的执行者之间进行传递和执行[1]。工作流技术的应用能够有效地提高企业业务流程的效率,进而提高企业的办公效率、生产效率,提高企业竞争力。

调度问题是在现实生活中普通存在的问题。工作流调度是找到一个满足业务逻辑约束的工作流任务执行的正确序列[2]。它与其它领域的调度问题相比,存在许多不同[3,4],其中最主要的是工作流是动态的,它具有较多的不确定性。这些不确定性包括工作流本身的不确定性,如工作流实例的到达是不确定的、工作流实例的执行路径和任务执行次数是不确定的;工作流运行环境的不确定性,如由于工作流执行过程持续较长的时间,可用的资源、数量和组成在这段时间里发生变化;资源的能力、任务的难度、资源当前的负荷、资源的兴趣等多个因素均会对任务的资源分配产生影响,而这些因素往往都是动态变化的。

现有的工作流调度方法可以分为两类:静态调度和动态调度。静态调度即事前调度,一次性地对工作流所包含的所有活动进行调度。这种调度方法理论上可以达到在当前状态下的全局最优。但随着工作流环境的动态变化、工作流实例的并发进行,静态调度的结果并不总是最优的。而且静态调度通常无法处理工作流的不确定性。目前已有的工作流静态调度方面的研究大都没有处理工作流的不确定性。如基于逻辑[5]、Petri 网[6]等的工作流调度,大多数是只考虑了满足时间和结构约束的工作流静态调度。Senkul 等人[2]提出的静态预调度方法考虑了工作流的资源约束,但没有考虑工作流的动态特性,也没有考虑系统中可能存在的多个并发工作流实例的情况。Baggio 等人[4]也采用静态调度的方法,他们考虑了工作流的不确定性,提出使用猜测的方式来处理工作流的不确定性。但采用这种猜测的方法,当实际情况与猜测情况产生偏差时,需要采取一系列的补救措施。

由于静态调度无法很好地处理工作流的动态性和不确定性,人们开始关注工作流的动态调度[7,8]。这些方法大多数是为每个待执行任务逐个分配资源。在调度时刻对于单个任务来说,其调度结果是最优的,但是很难达到全局最优。段永强等人[3]提出了一种基于资源约束的动态任务调度方法。该方法首先从当前待执行任务中挑选最优的可执行任务的集合,然后在考虑资源兴趣的情况下,根据资源能力和任务难度对任务进行分配。

本文将静态调度和动态调度两种方式结合起来,提出了

*)广东省自然科学基金(05200302)。肖志娇　博士研究生,主要从事工作流、资源管理、调度算法等研究;常会友　博士,教授,博士生导师,主要研究领域为协同系统理论与技术、工作流、信息系统集成。

一种工作流阶段性调度方法。该方法能够在妥善处理工作流的不确定性的同时，尽量使得任务调度在静态全局最优和动态单个最优之间取得均衡。本文的内容组织如下：第 2 节介绍本文提出的工作流阶段性调度；第 3 节通过仿真实验，分析了工作流阶段性调度的可行性和优越性；最后总结全文并提出未来的研究方向。

## 2 工作流阶段性调度

### 2.1 模型表示

Petri 网是一种很有效的模型表达方式。它不仅能描述系统的结构特性，同时能描述其动态特性。由于 Petri 网具有严密的形式化理论基础、基于状态的建模方式、丰富的分析技术[9]，Petri 网及其子类被广泛用于工作流领域[10~12]。

在工作流管理系统中，同时可能有同一个工作流的多个实例运行。基于传统 Petri 网所建立的工作流模型无法对它们进行区分。使用有色 Petri 网能够通过对托肯赋予不同的值来区分工作流的不同实例。Liu 等人[12]在工作流网[10]和有色 Petri 网[13]的基础上提出了有色工作流网的概念。由于在本文所采取的调度方法中，正确地表示不同工作流的不同实例的当前状态是非常重要的，因此使用有色工作流网来建模工作流。限于篇幅，有关有色 Petri 网、工作流网和有色工作流网的概念，可以参考文[13,10,12]，在此不再做一一介绍。

### 2.2 有色工作流网调度系统

为了便于接下来的描述，我们首先介绍多重集[14]的概念。

**定义 1**(多重集) 设 $S$ 为非空集合，$IN_0$ 是非负整数集，则从 $S$ 到 $IN_0$ 的函数叫做 $S$ 上的多重集。

多重集和集合的区别在于前者允许同一个元素出现多次。通常用 $S_{MS}$ 来表示集合 $S$ 上的所有有限多重集所组成之集合。设 $b\in S_{MS}$ 为 $S$ 上任一多重集，由定义可知，对任何 $s\in S$，$b(s)\in IN_0$，$b(s)$ 表示元素 $s$ 在 $b$ 中出现的次数。

为了方便工作流阶段性调度的进行，我们在使用有色工作流网描述的工作流模型的基础上建立有色工作流网阶段性调度系统。

**定义 2**(有色工作流网阶段性调度系统) $\Sigma=(D, P, T; A, C, G, E, I; SP, M_0, S_0)$是一个有色工作流网阶段性调度系统，当且仅当：

1) $WCPN=(D, P, T; A, C, G, E, I)$是一个有色工作流网[12]，其中 $D$ 是颜色集，$P$ 是库所集，$T$ 是变迁集，$A$ 是弧集，$C$ 是颜色函数，$G$ 是守卫函数，$E$ 是表达函数，$I$ 是初始化函数；

2) $SP\in P$ 是阶段调度点的集合，它包括开始节点库所、所有的“OR-Split” 库所，以及为了改进调度效率，而由用户指定的某些库所；

3) $M_0: P\to D_{MS}$，是$\Sigma$的初始执行标识，它满足条件：$\forall p\in P: M_0(p)\in C(p)_{MS}$；

4) $S_0: P\to D_{MS}$，是$\Sigma$的初始调度标识，它满足条件：$\forall p\in P: S_0(p)\in C(p)_{MS}$。

为了正确地表示工作流实例当前的调度状态，我们这里引入了调度标识的概念，以区别于表示工作流实例执行状态的标识。

**定义 3**(执行标识) $M: P\to D_{MS}$，是有色工作流网调度系统$\Sigma$的执行标识的条件是：$\forall p\in P: M(p)\in C(p)_{MS}$。

**定义 4**(调度标识) $S: P\to D_{MS}$，是有色工作流网调度系统$\Sigma$的调度标识的条件是：$\forall p\in P: S(p)\in C(p)_{MS}$。

### 2.3 工作流阶段性调度

工作流阶段性调度即根据所设置的调度点和工作流实例实际执行的情况，阶段性地对可调度的工作流任务进行调度。图 1 描述了对一个工作流实例进行阶段性调度的过程。其中的初始化步骤，除了要进行传统的初始化操作外，为了保证工作流阶段性调度的顺利进行，还要初始化阶段调度点集合 SP，并在开始节点库所中放置初始执行托肯和初始调度托肯。

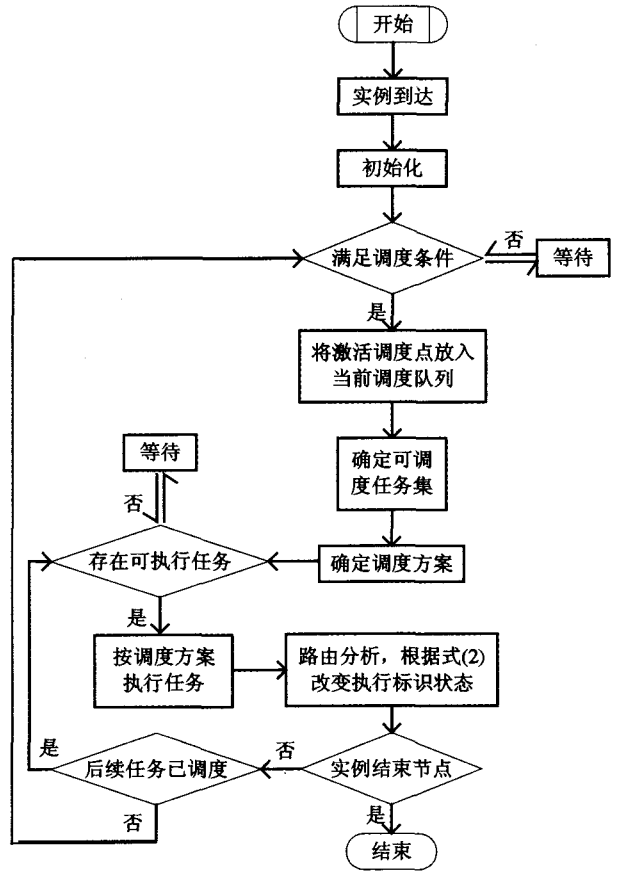


图 1 工作流阶段性调度流程图

接下来我们详细描述工作流阶段性调度的各个步骤。

#### 2.3.1 任务的调度

**定义 5**(可调度变迁) 变迁 $t\in T$ 称之为可调度变迁，当且仅当：$\exists c\in C(p), \forall p\in {}^\bullet t: S(p)(c)\geqslant 1$。其中，${}^\bullet t$ 是变迁 $t$ 的前驱，满足条件：${}^\bullet t=\{p|(p,t)\in A\}$。

**规则 1**(调度激发的条件) 当且仅当 $\exists sp\in SP, \exists c\in C(sp): S(sp)(c)\geqslant 1\wedge M(sp)(c)\geqslant 1$，且存在一个 $sp$ 的后续变迁为可调度变迁时激发调度。其中 $sp$ 称为激活调度点。

**规则 2**(调度标识产生规则) 可调度变迁 $t$ 调度激发后得到的后继调度标识 $S'(p)(c)$ 由下式给出：

$$S'(p)(c)=\begin{cases}S(p)(c)+1, p\in t^\bullet - {}^\bullet t\\ S(p)(c)-1, p\in {}^\bullet t - t^\bullet\\ S(p)(c), \text{其它}\end{cases} \quad (1)$$

其中，${}^\bullet t$ 是变迁 $t$ 的前驱，满足条件：${}^\bullet t=\{p|(p,t)\in A\}$；$t^\bullet$ 是变迁 $t$ 的后继，满足条件：$t^\bullet=\{p|(t,p)\in A\}$。

确定可调度任务集合的算法流程如图 2 所示。

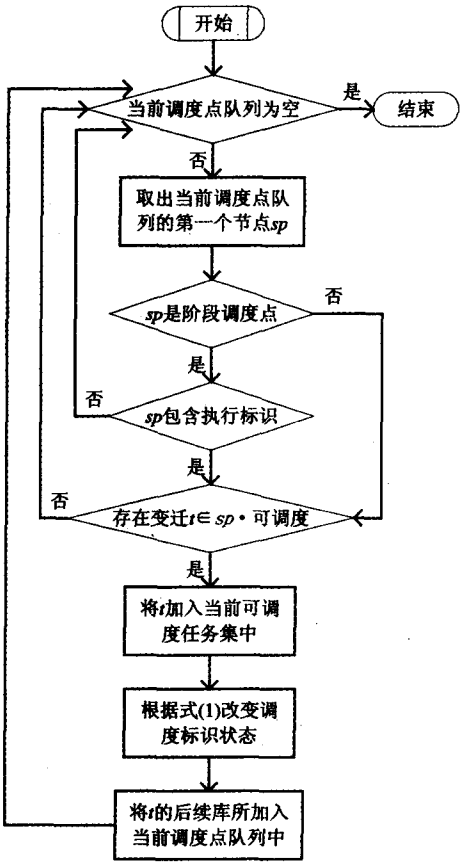


图 2　可调度任务确定算法流程图

2.3.2　调度方案的确定

工作流管理系统在当前阶段有 $m$ 个待调度的任务。系统中有 $m_i(i=1,\cdots,m)$ 个资源可以执行任务 $i$。调度方案的确定就是要从可执行任务 $i$ 的资源集合中选取合适的资源参与任务 $i$ 的执行，从而达到用户要求的相关性能指标。

该问题的求解，当问题规模不是很大时，可以采用精确的算法求解，如枚举法等。但在实际情况中，工作流系统中有多个工作流的多个实例同时运行，特别是当每个工作流的规模很大，结构复杂时，问题规模可以变得很大，采用精确的求解算法已经无法在可接受的时间内求解。这时可以采用启发式的方法，如模拟退火算法、遗传算法等求解。

2.3.3　任务的执行

可调度任务需要等待激活成可执行任务，才能被执行。

**定义 6**(可执行变迁)　变迁 $t\in T$ 称之为可执行变迁，当且仅当：变迁 $t$ 已被调度，且 $\exists c\in C(p),\forall p\in \cdot t: M(p)(c)\geqslant 1$。其中，$\cdot t$ 是变迁 $t$ 的前驱，满足条件：$\cdot t=\{p|(p,t)\in A\}$。

**规则 3**(执行标识产生规则)　可执行变迁 $t$ 执行后得到的后继执行标识 $M'(p)(c)$ 由下式给出：

$$M'(p)(c)=\begin{cases}M(p)(c)+1, p\in t\cdot - \cdot t\\ M(p)(c)-1, p\in \cdot t - t\cdot\\ M(p)(c), \text{其它}\end{cases}\qquad(2)$$

其中，$\cdot t$ 是变迁 $t$ 的前驱，满足条件：$\cdot t=\{p|(p,t)\in A\}$；$t\cdot$ 是变迁 $t$ 的后继，满足条件：$t\cdot=\{p|(t,p)\in A\}$。

2.3.4　支持工作流阶段性调度的工作流引擎

支持工作流阶段性调度的工作流引擎的主要架构如图 3 所示。

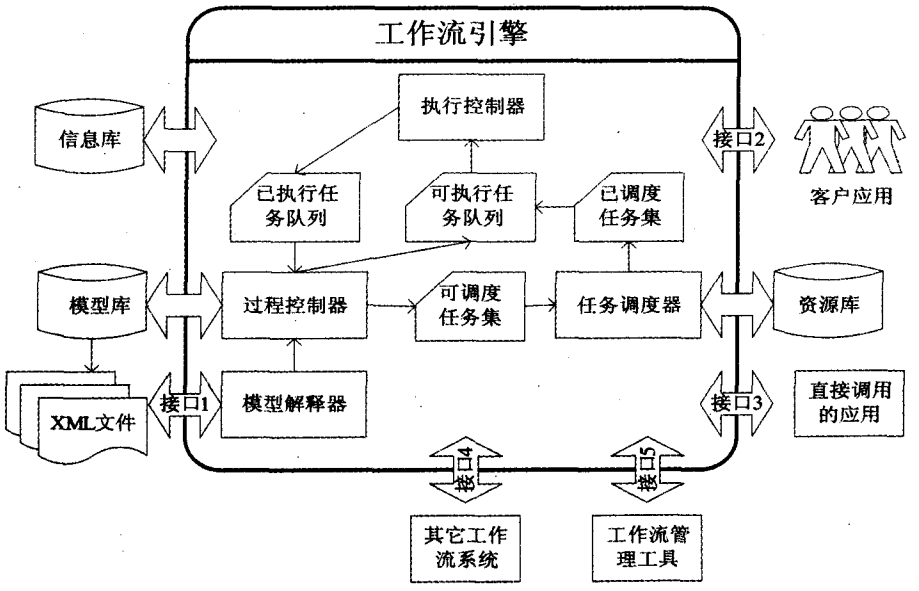


图 3　支持阶段性调度的工作流引擎架构图

系统使用 4 个容器来辅助实现工作流的阶段性调度的实现：

1) 可调度任务集：保存的是由可调度任务确定算法计算出来的可进行调度的任务。

2) 已调度任务集：保存已经进行了调度，但仍需等待激活的任务。

3) 可执行任务队列：保存已调度，并且已被激活的任务。

4) 已执行任务队列：存放已经执行完成的任务。

工作流引擎包含 4 个模块，各个模块的功能如下：

1) 模型解释器：用于解释工作流模型和其他业务模型并将工作流模型的全部或部分模型传递给过程控制器。它是工作流引擎的一个对外接口，引擎通过它获取模型。这里，我们使用 XML 语言描述工作流模型。采用 SAX 即时解析方式，从 XML 文件中读取模型信息，并将其转换成规定格式的容器中所存放的各种元素的类实例。

2) 过程控制器：工作流引擎的核心模块，负责流程实例

的管理与任务的导航。用户从客户端启动流程，系统便开一个过程控制器的线程全程跟踪管理，直至流程结束。过程控制器根据已执行任务分析流程路由，并根据已调度任务集中的调度结果，选择激活任务调度或激活任务为可执行任务。

3) 任务调度器：为任务指派合适的执行者和执行顺序。任务调度器为当前在可调度任务集中的所有可调度任务统一进行调度。这里，我们采用遗传算法求解调度方案。

4) 执行控制器：负责流程中任务的执行。主要的输入是从可执行任务队列取得的可以被执行的任务，输出的是已经完成的任务，被放入已执行任务队列。

## 3 仿真实验及结果分析

为了说明工作流阶段性调度方法的可行性和优越性，我们通过仿真实验，分别对使用静态调度、动态调度和本文提出的阶段性调度三种不同调度方法的工作流系统的性能进行了分析比较。其中静态调度采用文[4]中的方法来处理工作流路由的不确定性(假设猜测能够达到100%的准确率)，使用遗传算法来确定调度方案；动态调度采用文[15]中的方法。

工作流系统的目标就是尽可能快捷地完成任务[16]，任务拖期不仅会造成成本等方面的增加，还会带来许多意想不到的麻烦。因此，本文实验采用任务拖期概率(拖期任务的比例)作为评价指标。我们在使用不同调度方法的系统中，分别运行100个实例，它们是使用随机产生器产生的来自于5种不同工作流的实例。在运行的不同时期，我们抽取了10组数据进行比较，如图4所示。

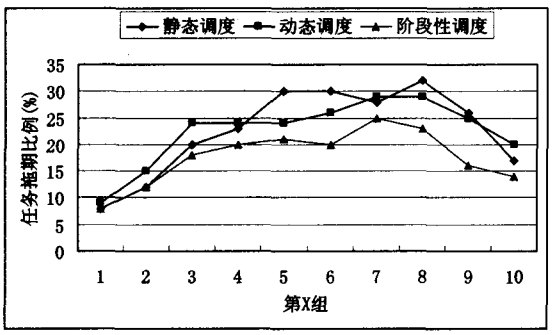


图4 不同运行时期不同调度方法下任务拖期概率的比较

通过图4可以看到，在工作流系统运行的初期，由于工作流任务较少，资源较充足，因此拖期任务较少；当工作流系统运行一段时间以后，拖期任务逐渐增加。在系统运行初期，不确定因素较少，工作流及其环境的变化较小。从图4中可以看出，此时静态调度的结果是较优的，阶段性调度也能取得较好的解，而动态调度的结果就不太理想。当系统变化频繁时，动态调度的结果较优，而静态调度却不能很好地适应动态变化，无法取得较好的解。而通过合理地设置调度点，阶段性调度所取得的解最好，有时甚至优于动态调度的解。

基于以上的仿真实验，我们可以得出如下结论：当工作流系统的变化较少时，使用静态调度更有利于提高系统的性能。但是静态调度不能有效地应对系统的动态特性。而且，在实验中我们假设所使用的静态调度对路由的猜测是100%正确。显然，在实际中这是不可能的。当工作流系统的环境变化频繁时，使用动态调度能达到更好的系统性能。动态调度能对各种动态变化做出灵敏的反应，使得每次调度对于单个任务来说总是最优的，但是却无法取得全局优化的效果。阶段性调度不需要对路由进行猜测，对动态环境的适应力也较强。它可以根据实际情况，在应对各类动态变化的同时，实现任务的成批预调度，从而取得一定的全局优化的效果。另外，阶段性调度可以通过少设置调度点来近似模拟静态调度，也可以通过在每个库所设置调度点来模拟动态调度。因此，在使用工作流阶段性调度时，可以根据实际情况，通过合理地添加阶段调度点，在动态调度和静态调度间取得一个较好的均衡。

**结束语** 静态调度的优点在于无须在任务执行时再费时查找合适的执行资源，而且在理论上可以达到在调度时刻系统状态下的全局最优。但随着工作流环境的动态变化，静态调度的结果并不总是最优的，而且它无法很好地处理工作流中的不确定性，通常使用一些猜测的方法来处理。而动态调度是在每个任务执行时再分配资源，能很好地处理工作流的动态性和不确定性，而且在调度时刻对于单个任务的调度是最优的，但很难达到全局最优。本文在总结这两种方法各自的优缺点的基础上，提出了一种基于有色Petri网的工作流阶段性调度方法。实验数据表明该方法能够在有效处理工作流动态不确定性的同时，使得任务调度在静态全局最优和动态单个最优之间取得一个较好的均衡。

如何根据实际情况，合理地甚至是动态地设置阶段调度点，从而在静态调度和动态调度间达到一个最佳的平衡是我们下一步的研究方向。另外，采用该方法进行工作流调度，当任务调度无法达到预期指标时，可以将已调度任务但尚未执行的任务重新加入调度集中进行调度。这样，本文问题就变成了一种可重入调度问题，对于这一类问题的研究也是我们下一步研究的方向。

## 参 考 文 献


1 Workflow Management Coalition. Terminology & Glossary [S]. WfMC-TC-1011. Workflow Management Coalition, 1999
2 Senkul P, Toroslu I H. An architecture for workflow scheduling under resource allocation constraints. Information System, 2005, 30: 399～422
3 段永强，曹健，张申生. 工作流系统中的动态任务调度. 中国机械工程，2002，13(3)：233～235，241
4 Tramontina G B, Wainer J, Ellis C. Applying scheduling techniques to minimize the number of late jobs in workflow systems. In: Proceedings of 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004,3: 1396～1403
5 Davulcu H, Kifer M, Ramakrishnan C R, et al. Logic based modeling and analysis of workflows. In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems, 1998. 25～33
6 Julia S, de Oliveira F F. A p-time hybrid Petri net model for the scheduling problem of workflow management systems. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2004, 5: 4947～4952
7 Shen Minxin, Tzeng Gwo-Hshiung, Liu Duen-Ren. Multi-criteria task assignment in workflow management systems. In: Proceedings of the 36th Hawaii International Conference on System Sciences, 2003. 9
8 Lee K M. Adaptive resource scheduling for workflows considering competence and preference. In: Negoita M Gh, et al. eds. KES 2004, LNAI 3214, Berlin, Heidelberg: Springer-Verlag, 2004. 723～730
9 van der Aalst W M P. Three Good Reasons for Using a Petri-net-based Workflow Management System [A]. In: Navathe S, Wakayama T, eds. Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC'96) [C]. Camebridge, Massachusetts, 1996, 11: 179～201
10 van der Aalst W M P. Workflow verification: finding control-flow errors using Petri-net-based techniques [A]. In: Business Process Management [C]. Lecture Notes in Computer Science, vol 1806. Berlin: Springer-Verlag, 2000. 161～183
11 Sea Ling, Schmidt H. Time Petri net for workflow modeling and analysis [A]. In: IEEE International Conference on Systems, Man, and Cybernetics [C], 2000,4:3039～3044
12 Liu Dongsheng, Wang Jianmin, Chan Stephen C F, et al. Modeling workflow process with colored Petri nets [J]. Computers in Industry, 2002, 12(3): 267～281
13 Jensen K. Coloured Petri Nets-Basic Concepts, Analysis Methods and Practical Use. 2nd ed. New York: Springer, 1997
14 袁崇义. Petri网原理. 电子工业出版社，1998
15 肖志娇，常会友，衣杨. 启发式规则与GA结合的优化方法求解工作流动态调度优化问题. 计算机科学，2006(已录用)
16 van der Aalst W M P, van Hee K. Workflow Management: Models, Methods, and Systems. MIT Press, 2002