

# 应用于 PLC 控制程序的 Petri 网执行模型

孟庆春 刘云卿

(中国科学院力学研究所技术发展部 北京 100190)

**摘要** 在现有的工业控制系统中经常使用 PLC 实现顺序控制、定时等功能。控制程序经常需要执行同步操作且输入开关量的触发时机具有不确定性,因此无法充分描述真实的执行过程。为此提出下述解决方法:首先在控制程序的编译阶段,使用 Petri 网建立网模型以表示程序执行逻辑,其次在控制程序的执行阶段,根据程序真实执行状况动态运行上述网模型,当程序运行结束时网模型停止执行,由此判断程序中存在的逻辑错误。

**关键词** Petri 网, PLC, 执行模型, 抑止弧

**中图分类号** TP393.08 **文献标识码** A

## Executive Model of Petri Net Applied in the PLC Control Program

MENG Qing-chun LIU Yur-qing

(Mechanical Institution, Chinese Academy of Sciences, Beijing 100190, China)

**Abstract** In existing industrial control system, PLC (Programmable Logic Controller) is often applied to realize sequential control, timing and etc. For the reason that control program often performs synchronous control and the occasions of input switches triggered are undermined, it is inevitably to cause the problem that control program cannot describe real executing process of control program fully. In view of it this paper put forward a method as follows: Firstly, we can use Petri net to construct net model representing the control logic during the process of compiling control program, this model represents executing logic of the PLC program. Secondly, we can dynamically run the net model mentioned above according to the actual execution route of the control program until the program stops executing. So we can find out some of the logical errors.

**Keywords** Petri net, PLC, Execution model, Prohibit arc

## 1 引言

传统工业控制系统中普遍采用继电器控制,但继电器控制方式存在控制系统体积大、耗电多、寿命短、可靠性差及运行速度慢等缺点。为了克服上述缺点,1969 年美国 digital 公司研制出第一代可编程序控制器 (PLC) 以代替继电器控制设备<sup>[1]</sup>。PLC 问世以来,很快被应用到机械制造、冶金、矿业、轻工等各个领域,大大推进了机电一体化进程,因此被人们称为现代工业控制三大支柱之一。

由于 PLC 主要用于实现顺序控制、定时计数等控制功能<sup>[2]</sup>,因此研究人员首先设计了类似继电器线路的梯形图以表达控制逻辑。为进一步简化梯形图的表达方式,人们编制了控制算法功能块(或子程序)作为指令系统,并将其固化在存储器 ROM 中,这样用户就可以通过编制类似汇编语言的程序来实现控制逻辑。根据 1993 年 IEC 制订的可编程序控制器标准 IEC61131-3 的规定,可编程序控制器的编程方式有 5 种<sup>[3]</sup>:指令表(IL)、梯形图(LD)、顺序功能图(SFC)、功能块图(FBD)、结构化文本(ST),其中梯形图和结构化文本得到了广泛的应用。

虽然梯形图和结构化文本为用户提供了方便的编程接口,但是由于控制程序经常需要执行同步操作,因此控制程序中的不同部分具有一定的逻辑执行顺序,即使用户按照上述顺序编制程序,但是由于输入开关触发时机具有不确定性,因此程序的真实执行顺序与程序编制时采用的顺序仍有可能并不相同,所以控制程序无法充分描述真实的执行过程。为了解决上述问题,本文引入了 Petri 网技术。

Petri 网技术是由 C. A. Petri 于 1962 年在名为“用自动机通信”的博士论文中首先提出的<sup>[4]</sup>。20 世纪 60 年代末至 70 年代初,美国 ADR 公司信息系统理论组的 A. W. Holt 进一步丰富和发展了有关 Petri 网的理论、记号及表示法,同时,MIT 的 J. B. Dennis 教授领导的计算机结构课题组也在 Petri 网理论方面进行了许多研究工作。随着 Petri 网技术研究的深入,并发现现象逐渐得到研究人员的重视,因而 Petri 网技术成为计算机领域和自动化领域的热门研究课题。

Petri 网本身是一种描述同步并发现象的系统模型,它既有严格的数学定义,又有直观的图形表示,因此成为一种有效描述并发系统行为的技术。本文所述的 PLC 控制程序执行过程虽然本质上属于同步操作,但是由于输入开关量触发时

到稿日期:2008-11-24 返修日期:2009-03-03 本文受中科院所技术开发专项基金(国科计字[2000]056号)资助。

孟庆春(1979-),男,博士,助理研究员,主要研究方向为机动车检测与网络技术, E-mail:mqc\_794388@hotmail.com;刘云卿(1949-),男,研究员,主要研究方向为机动车检测与网络技术。

机的不确定,因此程序执行类似一种同步并发过程,故文中采用 Petri 网建立 PLC 控制程序的执行模型。

由于 PLC 控制程序需要完成顺序控制、定时计数等功能,因此本文选用带有抑制弧的时间 Petri 网模型描述 PLC 控制程序,建立和执行过程如下:在 PLC 控制程序的编译阶段建立代表控制逻辑的 Petri 网模型,该模型描述了控制程序开关量之间基于输入输出依赖的同步关系和带有时序约束的并发关系。在 PLC 控制程序的运行阶段,动态执行 Petri 网模型,控制程序运行结束网模型停止执行。

利用 Petri 网技术建立的 PLC 控制程序执行模型不但可以充分描述程序的执行过程,而且还可以在程序执行前检测执行模型的活性、互斥性等性质,因而可以在程序执行前预先判断出程序存在的部分逻辑错误,从而减少了调试时间。

## 2 Petri 网

Petri 网是一种图形化的建模工具,它能较好地描述离散事件的动态过程,并能精确描述事件的顺序、并发和冲突关系,其模拟能力与图灵机等价<sup>[5,6]</sup>。目前 Petri 网已经被广泛应用于线路设计、网络协议、软件工程、人工智能、形式语义、操作系统、并行编译等诸多领域。下面首先给出 Petri 网技术涉及的基本概念,然后针对 PLC 控制程序建模应用的具体技术展开介绍。

### 2.1 基本概念

**定义 1**(资源,resource) 是在系统发生变化所涉及的与系统状态有关的因素,包括原料、部件、产品、人员、工具、设备、数据和信息。

**定义 2**(库所,place) 是存放资源的场所,既表示一个场所,又表示该场所存放有一定的资源,用符号  $P$  表示库所。库所又称为状态元素,即  $P$  元素。

**定义 3**(变迁,transition) 是资源消耗、使用和产生的一种变化。用符号  $T$  表示变迁。变迁又称为  $T$  元素。

**定义 4** 网是一个三元组  $N = (P, T, F)$ ,其中

- 1)  $P$  是库所集合;
- 2)  $T$  是变迁集合,且与  $P$  不相交;
- 3)  $F$  是流关系,  $F \subseteq (P \times T) \cup (T \times P)$ ,即连接弧集合;
- 4)  $P \cap T = \emptyset, P \cup T = N$ ,网  $N$  中的元素;
- 5)  $dom(F) = cod(F) = P \cup T, dom(F) = \{x | y : (x, y) \in F\}, cod(F) = \{x | y : (y, x) \in F\}$ 。

**定义 5** Petri 网是一个六元组  $PN = (P, T, F, K, W, M_0)$

- 1)  $N = (P, T, F)$  是基本网的定义;
- 2)  $K$  是  $N$  上的容量函数;
- 3)  $W$  是  $N$  上的容量函数;
- 4)  $M_0$  是  $N$  上的初始标识。

**定义 6** Petri 网的图形表示如下:  $P = (p_1, p_2, p_3, \dots, p_n)$  表示库所的集合,用  $\circ$  表示,  $T = (t_1, t_2, \dots, t_m)$  表示变迁的集合,用  $\square$  表示,  $F$  表示流关系,用  $\rightarrow$  表示。

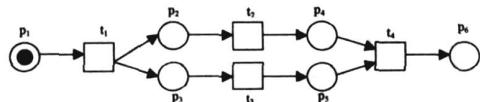


图 1 Petri 网图

如图 1 所示,  $P = (p_1, p_2, p_3, p_4, p_5, p_6)$ ,  $T = (t_1, t_2, t_3, t_4, t_5)$ ,  $p_1$  为初始状态。

Petri 网在发展过程中为了满足不同的应用需求,不断融入新的表示方法,在基本网的基础上发展出抑制弧 Petri 网、谓词/变迁 Petri 网、时间 Petri 网、时序 Petri 网、染色 Petri 网、面向对象 Petri 网、随机 Petri 网、被控 Petri 网、混合 Petri 网等各类网模型。为了有效描述 PLC 控制程序,本文使用带有抑制弧的时间 Petri 网模型。

Petri 网中引入了抑制弧来扩展 Petri 网的描述能力<sup>[7]</sup>:增加 Petri 网中变迁节点对输入库所的零检测能力,当输入库所不含有托肯时促使变迁发生。

### 2.2 时间 Petri 网

时间 Petri 网可以表示为一个六元组  $TPN = (P, T, F, M_0, In, Out)$ <sup>[8]</sup>,其中  $P, T, F, M_0$  满足基本 Petri 网中的定义,  $In$  和  $Out$  表示  $T \times P \rightarrow T \times P$  的函数。若  $In, Out$  定义为:  $In(t, p), Out(t, p)$ , 则库所  $p$  即可能是变迁  $t$  的输入库所也可能是输出库所。标记  $M$  是一个定义在  $P \times R^+$  ( $R^+$  为非负实数集)上的多重集,其包含每个库所的托肯数量和引发时延,引发时延表示变迁满足引发条件的时间点至变迁发生的时间间隔。

## 3 Petri 网模型建立和执行

PLC 控制程序在编译过程中经过词法和语法分析进入语义分析,语义分析则被进一步划分为两个阶段:首先进行与 PLC 语言相关的语义分析;其次利用 Petri 网技术建立 PLC 控制程序执行模型,该模型描述了程序执行过程中的同步和并发关系。当控制程序真实运行时,动态执行上述模型,至控制程序结束网模型执行完毕。

### 3.1 模型建立

在 PLC 控制程序中,PLC 控制器的某些 IO 端口之间具有基于输入输出依赖的同步关系,而其它端口并不具有这种约束,外部信号到达这些端口的时序虽然是固定的,但是具体时间却不确定,因此那些不具有同步关系的 IO 端口之间具有受到时序约束的并发关系。

为了有效描述 IO 端口之间的同步并发关系及 PLC 控制程序自身具有的与或逻辑关系,需要建立适当的 Petri 网模型,因此本文采用带有抑制弧的时间 Petri 网,在上述网模型中通过抑制弧对 IO 端口进行零测试,通过引发时延表示定时器的定时操作,通过引入  $and\_join$  和  $or\_join$  操作<sup>[9]</sup>表示 PLC 控制程序的与或逻辑关系。

如图 2 所示,  $P_1, P_2$  表示 IO 端口  $X_1, X_2$  处于待输入状态,这两个端口之间不具有同步关系,可以并发接收外部信号的输入,因此作为初始态有托肯的独立库所出现在图中,变迁  $t_1$  表明控制程序中端口  $X_1, X_2$  输入的数值进行逻辑或操作且端口输入应该在 1.5s 的时限内完成,以保证满足控制器的执行时序,  $P_3$  表示 IO 端口  $Y_1$  处于输出状态,该端口对外输出的数值是端口  $X_1, X_2$  输入数值逻辑或操作的结果,因此三者之间具有同步关系。

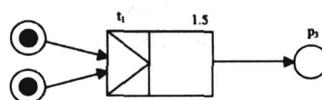


图 2 控制程序 Petri 网模型示意图

### 3.2 模型执行

模型执行过程虽然遵循 PLC 控制器的时序,但是受到外界信号引发时间的约束,因此同一程序在不同的外部信号输入条件下执行过程可能不同,具体算法如下所述:

1) 首先通过硬件设备检测外部信号输入,如常开触点闭合、定时器复位,然后减少对应库所  $p$  的托肯计数(默认状态下减 1,如果与库所  $p$  相连的输出弧上有对应的计数  $n(n > 1)$ ,则减去数值  $n$ ),如果库所  $p$  的托肯计数减为零,则将其置于激活状态;

2) 如果某变迁  $t$  的所有输入库所都处于激活状态且满足时序约束,则将变迁  $t$  置于激活状态,并将对应的所有输入库所置于常态;

3) 如果某变迁  $t$  处于激活状态且有输出库所  $p$  与其相连,则库所  $p$  的托肯计数增加(默认状态下加 1,如果与变迁  $t$  相连的输出弧上有对应的计数  $n(n > 1)$ ,则增加数值  $n$ ),如果与库所  $p$  相连的所有输入变迁都处于激活状态,将库所  $p$  置于激活状态,并将对应的所有输入变迁置于常态;

4) 重复执行步骤 1 - 3 的过程,直到 PLC 控制程序运行结束,网模型停止执行。

### 4 建模范例

本节将给出一个 PLC 程序执行电动机正反转控制的具体例子,使用 Petri 网技术建立执行模型并说明模型执行过程。

某些生产机械(如吊车)的提升机构需作上下和左右两个方向的运动,这就要求产生运动的电动机必须能作正、反两个方向的旋转。由异步电动机的工作原理可知,要使电动机反向旋转,需对调三根交流电源线中的两根以改变定子电流的相序,因此实现电动机的正反转需要两组启动开关,其电路图如图 3 所示。

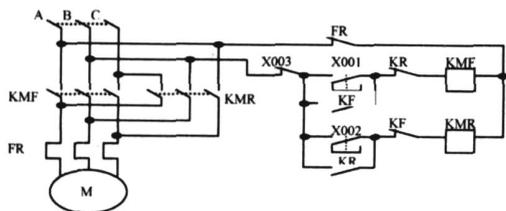


图 3 电动机正反转控制电路图

图 3 中左半部分为主电路,其余为控制电路。在主电路中,若正转启动开关 KMF 闭合,则电动机 M 正转,若 KMF 断开而反转启动开关 KMR 闭合,则电动机的三根交流电源线中有两根位置对调,因此电动机反转。若开关 KMF 和 KMR 同时闭合,将造成电源短路。若控制电路中的停止开关 FR 断开,电动机停止运动。在控制电路中 X003 表示停止开关, X001 表示正转启动开关, X002 表示反转启动开关, X003 表示停止开关。为了与 PLC 控制程序编写标准保持一致,本例采用 Y001 表示正转线圈及线圈保持控制开关 KF, Y002 表示反转线圈及线圈保持控制开关 KR。

根据上面分析的控制功能,图 3 中的控制电路可以用梯形图表示,如图 4 所示。

将图 4 所示的梯形图程序表示为 Petri 网模型可以得到图 5。

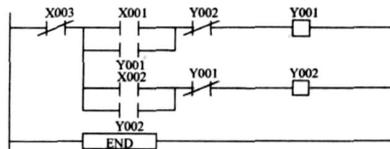


图 4 电动机正反转控制程序梯形图

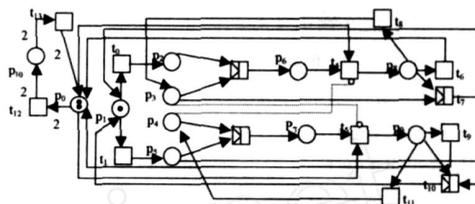


图 5 电动机正反转控制程序 Petri 网模型图

库所  $p_0$  表示停止按钮处于常闭状态,库所  $p_1$  表示正反转启动按钮处于常开状态,库所  $p_2$  表示正转启动按钮处于闭合状态,库所  $p_3$  表示正转线圈保持开关处于常开状态,库所  $p_4$  表示反转线圈保持开关处于常开状态,库所  $p_5$  表示反转启动按钮处于闭合状态,库所  $p_6$  表示逻辑或程序块处于输出状态,库所  $p_7$  表示反转线圈开关处于常开状态,库所  $p_8$  表示正转线圈处于充电输出状态,库所  $p_9$  表示反转线圈处于充电输出状态,库所  $p_{10}$  表示停止按钮处于断开状态。

变迁  $t_0$  表示正转启动按钮闭合,变迁  $t_1$  表示反转启动按钮闭合,变迁  $t_2$  表示逻辑或程序块执行逻辑运算,变迁  $t_3$  表示逻辑或程序块执行逻辑运算,变迁  $t_4$  表示正转线圈充电,变迁  $t_5$  表示反转线圈充电,变迁  $t_6$  表示电动机正转条件下停止按钮保持常闭状态,变迁  $t_7$  表示正转启动按钮复位,变迁  $t_8$  表示正转线圈开关复位,变迁  $t_9$  表示电动机反转条件下停止按钮保持常闭状态,变迁  $t_{10}$  表示反转启动按钮复位,变迁  $t_{11}$  表示反转线圈保持开关复位,变迁  $t_{12}$  表示停止按钮断开,变迁  $t_{13}$  表示停止按钮复位。

当正转启动按钮 X001 按下时,库所  $p_1$  失去托肯经过变迁  $t_0$  到达库所  $p_2$  ( $p_2$  获得托肯),初始状态下由于库所  $p_3$  不含托肯,两者在变迁  $t_2$  经过逻辑或运算对外输出,这样库所  $p_2$  失去托肯,同时库所  $p_6$  获得托肯,当停止按钮处于常闭状态(库所  $p_0$  具有托肯)且反转线圈保持开关处于常开状态(库所  $p_4$  不具有托肯),经过变迁  $t_4$  到达库所  $p_8$  ( $p_1$  失去一个托肯, $p_6$  失去托肯, $p_8$  获得两个托肯),正转线圈通电,电动机正转对外输出,正转线圈通电后经过变迁  $t_6$  恢复库所  $p_0$  失去的一个托肯,同时通过  $t_7$  使正转线圈保持开关闭合(库所  $p_3$  获得托肯),通过变迁  $t_6$  和  $t_7$  库所  $p_8$  消耗了两个托肯,激活状态转向库所  $p_0$  和  $p_3$ ,由于此时库所  $p_2$  不具有托肯,因此表示正转启动按钮处于断开状态,但是由于正转线圈保持开关处于闭合状态(库所  $p_3$  具有托肯),因此可以通过变迁  $t_2$  到达库所  $p_6$ ,同时由于库所  $p_0$  具有托肯,因此可以通过变迁  $t_4$  到达库所  $p_8$ ,保持电动机正转对外输出。电动机的反转过程与上同理可以得到。

当电动机正转过程中希望切换到反转过程时,由于库所  $p_8$  或者  $p_3$  具有托肯,因此通过逻辑或操作经变迁  $t_7$  到达库所  $p_1$  ( $p_1$  获得托肯),按钮 KMF 和 KMR 都处于常开状态,当反转按钮 KMR 闭合时,库所  $p_1$  经变迁  $t_1$  到达库所  $p_5$ ,反转过程开始。同理可以得到电动机从反转过程切换到正反转过程。

(下转第 159 页)

- [25] Vladimir V K. Mathematical Methods in Queuing Theory [M]. Kluwer Academic Publishers, 1994
- [26] Finch P D. Balking in the queueing system GI/M/1 [J]. Acta Math. Acad. Sci. Hungar., 1959, 10:241-247
- [27] Sigman K. Appendix: A Primer on Heavy-tailed Distributions [J]. Queueing Systems, 1999, 33(1-3):261-275
- [28] Charles M, Claudia K. Subexponential Distributions [M]. J. A. Robert, et al., eds. A practical guide to heavy tails: statistical techniques and applications. Boston: Birkhauser, 1998
- [29] Takacs L. On a Combined Waiting Time and Loss Problem Concerning Telephone Traffic [J]. Ann. Univ. Sci. Budapest. Eotvos. Sect. Math., 1958, 1:73-82
- [30] Shortle J F, Brill P H, Fischer M J. An Algorithm to Compute the Waiting Time Distribution for the M/G/1 Queue [J]. INFORMS Journal on Computing, 2004, 16(2):152-161
- [31] Piessens R, Huysmans R. Algorithm 619 Automatic Numerical Inversion of the Laplace Transform [J]. ACM Trans. on Mathematical Software, 1984, 10(3):348-353
- [32] Abate J. Numerical Inversion of Laplace Transforms of Probability Distribution [J]. ORSA Journal on Computing, 1995, 7(1):36-43
- [33] 汪浩. 一个具有随机丢弃分组机制且分组成批到达的 GI<sup>X</sup>/M/1/N 排队系统[J]. 数学的实践与认识, 2005, 35(9):113-120
- [34] 汪浩, 李晓明, 严伟. 一类具有随机离去策略的排队系统及其在 Internet 拥塞控制中的应用[J]. 计算机科学, 2007, 34(12):37-40

(上接第 152 页)

当电动机处于正转过程中, 停止按钮按下, 库所  $p_0$  通过变迁  $t_{12}$  到达库所  $p_{10}$  (库所  $p_0$  消耗两个托肯, 库所  $p_{10}$  获得两个托肯), 电动机正转停止; 电动机反转过程中按下停止按钮与上同理可得。当停止按钮重新按下时, 库所  $p_{10}$  通过变迁  $t_{13}$  到达库所  $p_0$  (库所  $p_{10}$  消耗两个托肯, 库所  $p_0$  获得两个托肯)。

## 5 模型分析

使用 Petri 网作为建模工具时, 需要考虑的基本性质有<sup>[10]</sup>:

1. 有界性, 刻画状态空间的有限性;
2. 活性, 与所有可达标识下的潜在发生权有关。无死锁是一个弱化的条件, 它指网系统模型的全局的无限动作能力 (即动作发生), 但它的某个部分可能根本不工作;
3. 可逆性, 刻画从任何可达的标识到初始标识的可恢复性;
4. 互斥性, 刻画子标识不可能同时成立或变迁不可能并发执行的特征。

下面以第 3 节中的电动机正反转控制模型为例说明上述性质。

1. 前述网模型中具有标识为  $p_0 - p_{10}$  的 11 个表示系统状态的库所节点, 因此状态空间大小有限, 有界性条件满足;

2. 网模型中从初始态 (库所  $p_0$  具有两个托肯, 库所  $p_1$  具有一个托肯), 根据外界开关量的输入可以实现电动机正转、反转和停止运动三种不同的运动方式, 并且在任一状态只要存在开关量的输入网模型便可以无限运行下去, 因此具有活性特征;

3. 在初始态下, 库所  $p_0$  具有两个托肯, 库所  $p_1$  具有一个托肯, 在电动机正转的情况下, 操作员若希望切换到反转状态, 首先就要断开正转启动按钮, 这时网模型中库所  $p_8$  通过变迁  $t_6$  和  $t_7$  恢复库所  $p_0$  和  $p_1$  中消耗的托肯回到到初始状态, 同理电动机从反转切换到正转时也要经过类似的过程, 因此系统的可逆性在程序逻辑允许的情况下得到满足;

4. 当正转启动按钮被操作员按下时, 网模型中库所  $p_1$  通过变迁  $t_0$  到达库所  $p_2$ , 库所  $p_1$  中的托肯消耗为零, 在这种条件下库所  $p_1$  不可能通过变迁  $t_1$  到达库所  $p_5$ , 因此避免了电动机正反转开关同时闭合的可能, 系统互斥性根据控制程序逻辑得到满足。

**结束语** 本文利用 Petri 网技术建立了 PLC 控制程序的执行模型。在 PLC 控制程序的运行过程中动态执行 Petri 网模型, 当网模型执行完毕时控制程序结束运行, 这样如果编程人员或用户需要了解控制程序的具体执行过程, 可以将网模型的执行过程记录在指定的文件中, 从而满足程序使用者的特定需求。另外在控制程序执行前通过检测 Petri 网模型具有的活性、互斥性等性质, 可以发现部分逻辑错误以减少调试时间。虽然该执行模型具有上述优点, 但会增加控制程序执行所需的时间, 如果控制程序属于严格实时的任务, 必须最大限度减少执行 Petri 网模型所需的时间, 因此如何优化 Petri 网模型以适应实时的要求是今后进一步的研究方向。

## 参考文献

- [1] Yang Gong-yuan. Theory and Application of Programmable Logic Controller [M]. Beijing: Publishing House of Electronic and Industrial, 2004
- [2] Yan Ying-fu. Foundation of Programmable Logic Controller [M]. Beijing: Publishing House of People's Postal, 2005
- [3] Xie Ke-ming. Theory and Program Design of Programmable Logic Controller [M]. Beijing: Publishing House of Electronic and Industrial, 2003
- [4] Petri C A. Communication with automata [D]. University of Bonn, Bonn, 1962
- [5] Reisig W. Petri nets, an introduction [A]. Brauer W, Rozenberg G, Salomaa A, eds. EATCS, Monograph on Theoretical Computer Science [C]. Berlin: Springer Verlag, 1985
- [6] Zurawski R, Zhou Mengchu. Petri nets and industrial applications: a tutorial [J]. IEEE Transaction on Industrial Electronics, 1994, 41(6):567-583
- [7] Yuan Chong-yi. Theory and Application of Petri Net [M]. Beijing: Publishing House of Electronic and Industrial, 2005
- [8] Zaitsev D A. Invariant of timed petri nets [J]. Cybernetics and Systems Analysis, 2004, 40(2):226-237
- [9] van Aalst W, van Hee K. Workflow management models, methods and systems [M]. MIT Press, 2002
- [10] Grault C, Valk R. Petri Net of System Engineering [M]. Beijing: Publishing House of Electronic and Industrial, 2005