# 第四章 运用Petri网络的工作流建模

## 第一节Petri网络概述

Petri网络，中文通常翻译为裴氏网或者派翠网，是对并行离散系统的一种数学表示。这个方法在20世纪60年代由德国数学家和信息学家Carl Adam Petri发明，用来描述异步的、并发的工作流程系统模型。当时发明Petri网的出发点主要是受到相对论的启发，从数学模型的角度去描述物理世界的并发现象。据Petri教授本人说，他认为60年代自动化模型由于缺乏并发的概念不适合于表达现代物理学理论，例如爱因斯坦的狭义相对论和量子力学中的不确定性原理。

经过过去几十年的发展，Petri网络从一种简单的并发模型发展，开始被应用于各个不同的领域进行系统的建模、分析和控制。如对互联网分析性能、人工智能推理、神经网络、并行程序的设计、信号的处理和验证、工业制造控制系统等。特别发现了Petri网可以用于AI人工智能的推理和形式化模型的建立，可以表达流程的各种部件，例如顺序、并行、选择、循环、互斥和交替关系等。随着计算机和信息技术的告诉发展，人们发现在软件工程中，Petri网可以用于软件系统的分析和建模，比如将带颜色的Petri网，用于大型软件系统的说明、设计、仿真和实现，在软件开发生命周期的各个阶段，使用Petri网络都可以进行很好的指导。

Petri网拥有一套适用于多种系统的图形化、数学化的建模工具，为描述和研究具有异步、并行、分布和随机等特征的复杂系统提供了强有力的手段。Petri从狭义相对论出发，旨在描述各个事件和变化之间的因果关系，并从此出发，构建了“时序”，确立了了以下Petri网络的指导思想：如果存在两个变迁都被激活了，则它们都有可能发生。不能认为其发生的先后有任何的关系。两个时间和空间点之间，如果两者之间没有依赖关系或者因果关系，它们就是互相独立的。我们不能说这个发生在那个之前，或者那个发生在这个之前。

本文所关注的工作流程是近年来发展最快的信息应用之一。按照工作流管理联盟的定义，“工作流管理的关键在于业务流程的全电子化”。工作流管理为一个古老的问题提供了一个全新的解决方案，这个问题就是如何控制、监控、优化和支持业务流程。为了使用计算机能够处理工作流，就必须需要将业务流程的模型逻辑化，用图形和数学表达出来。Petri网正是这样一个建模工具建立和分析的过程。一方面，Petri网可以用作一个设计语言可以作为复杂工作流程的规范描述语言。另一方面，Petri网络理论的发展也提供了被验证有效的强大工具，可以被用来分析工作流程序的正确性。作为一种图形化工具，可以把Petri网看作和流程图类似的方法。作为一种数学化工具，它能够建立起状态方程、代数方程和其他描述系统行为的数学模型。

上一章已经提到，和其他的工作流建模方法比较起来，Petri网络的优势是既有直观的图形表达方式，又有形式化的数学表述方式；既有对实际系统描述的诸多手段，又有以计算机科学分析技术为坚实基础的概念描述。Petri网络拥有能够恰当处理因果上的不存依赖性的并发现象和表示不确定性选择的能力，所以目前产业界和研究界倾向于认为，Petri网是所有流程定义语言之母。

## 第二节Petri网络构造

从名字就可以知道，Petri网络是一种网络状的结构模型，或者是图（graph）模型，同时其模型又有动态的运行行为，满足一个自动机的特征。Petri模型可以用图形表示的方法，为简单的过程建立模型。图形由库所（Place）和变迁（Transition），有向弧（Arc），和令牌（Token）等基本元素组成的。我们可以将任何一个四元组（库所，变迁，输入函数，输出函数）映射到这个图形中去，也可以将这样的图形映射到这样一个四元组上。关于这四者的关系，简单地说，“库所表示系统的状态，变迁表示资源的消耗、使用及使系统状态产生的变化。变迁的发生受到系统状态的控制，即变迁发生的前置条件必须满足；变迁发生后，某些前置条件不再满足，而某些后置条件则得到满足。另外，库所中令牌决定了变迁的激活（Enabled）和激发（Fire），变迁的激发又将改变令牌的分布。以变迁激发导致令牌在库所之间的流动。”

### Petri网的图形结构

* 库所（Place）是圆形节点
* 变迁（Transition）是方形节点
* 有向弧（Arc）是库所和变迁之间的有向弧
* 令牌（Token）是库所中的动态对象，用黑色圆点表示，可以从一个库所移动到另一个库所；值得注意的是，令牌不能存在于变迁之中。

一个简单的Petri 图形表达如下：

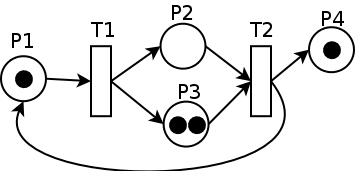


图 4-1 简单Petri 图形

### Petri网络的规则

* 库所和变迁是网络中的两类节点。
* 库所和变迁之间的弧是有方向的。
* 有向弧上可能有数字表示每次需要转移的令牌数量，默认为1。
* 两个库所或变迁之间不允许有弧。
* 库所可以拥有任意数量的令牌，或者没有令牌，有些网络中库所有令牌数量上限或者下限。
* Petri网络的状态是由标志函数（Marking）决定的，代表了令牌在库所之间的分布。
* 库所含有一个以上的令牌，则该库所被激活(Enable)。
* 若一个变迁的所有前置库所都被激活，则该变迁被激活。
* 一个被激活的变迁可以激发（Fire），也可以不被激发。激发的。
* 当变迁激发时，前置库所的令牌被消耗，后置库所的令牌被生成。
* 变迁激发是原子的（Atomic）和不确定的（un-deterministic），其顺序和时间都没有定义。

### Petri网的形式化表达

使用数学语言将上述描述表达，Petri网络可以用一个四元组表示

满足以下条件：

1. P是库所Places的有限集合
2. T是变迁Transitions的有限集合
3. 表示库所和变迁是两类不同的元素
4. 表示库所和变迁中至少有一个元素
5. 表示Petri网络中的流关系，其中的‘×’表示*笛卡儿积*
6. 表示输入元素或者前置库所集合（Preset）
7. 表示输出元素或者后置库所集合（Postset）
8. 一个标识（marking）是一个从P到自然数的函数，代表了每一个库所包含令牌的数量。
9. 激活，当且仅当
10. 定义权重函数，对于所有的

t激发前后的标识满足

而Petri网络系统则可以用来表示，其中称为初始标识。

### Petri网络用于建模

为了建模工作流，我们可以从条件事件网络（Event-Condition Net）得到启发。在建模过程中，我们通常会套用条件和事件的概念，变迁代表事件，库所就代表条件。当事件的前置条件成立时，这个事件就被触发，然后通常情况下，那个事件的前置条件就不再成立，它的后置事件被激活了，处于成立状态。某一个事件有一定数量的发生条件，即输入库所，事件的结果是触发了其他一些条件，即输出库所。网络中库所包含的令牌则代表某个条件发生时可供使用的资源或者数据。

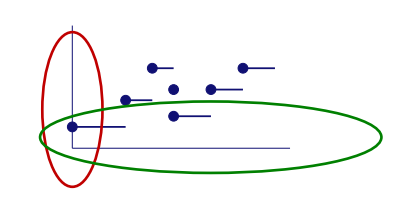


图 4-2 过程建模和数学建模

我们通常比较熟悉系统的数据设计建模。系统建模其实等于数据建模加上过程建模。如上图中红色轴的建模代表了数据建模，是业界研究的比较多的课题，对于绿色过程建模则达不到这种水平。事实上，我们可以为离散动态系统建立以下的模型：

1. S是有限的状态空间，由状态组成
2. 代表了状态间迁移的关系
3. 代表了初始状态

显然可得，Petri网定义了一个动态模型：

在实践中通常将以下建模为库所：

* 通信媒介，例如电话线、中间人和通信网络
* 缓冲区，例如队列、存储器和回收站
* 地理位置，例如仓库位置、办公室或医院
* 一个可能的状态条件，例如电梯所在的楼层

以下通常被建模为变迁：

* 事件，例如开始工作、病人去世、四季变化和信号灯切换
* 物体的变换，例如维修产品、更新数据库和维护文档
* 物体的传输，例如物流和文件传输

通常将以下物体建模为令牌：

* 实际物体，例如产品、部件、材料或者人员
* 信息实体，例如消息、信号或报表
* 一系列物体，例如一批产品、仓库中的部件和地址汇总
* 状态指示，例如信号灯或者过程中的状态界面
* 一系列状态指示用于表示某个条件已经被满足

### Petri网络的系统性质研究

* 变迁的是原子性的，也就是说，没有发生了一半的变迁，变迁不是发生了就是没有发生。
* 如果变迁的输入库所的数量与输出库所的数量不相等，网络中令牌的总数将发生变化，换句话说令牌数目不守恒。
* 当两个或者多个变迁被激活时，只能至多有一个变迁被激发。如果几个变迁同时激发的话，它们对库所中的令牌或产生竞争条件，不便于分析，所以一次只能发生一个变迁。
* Petri网络本身的结构是是静态的，也就是说，不存在发生了变迁生成了另一个变迁或者库所，从而改变Petri网结构的可能性。这方便了静态分析，也是Petri网络的局限之一。
* Petri网的状态由令牌在库所的分布决定。也就是说，变迁发生完毕、下一个变迁等待发生的时候才有确定的状态，正在发生变迁的时候是没有一个确定的状态的。

## 第三节 Petri网络用于工作流建模

上一章提到，对于工作流的理论上最优化的模型并不存在。我们往往发现难以衡量建模的质量，或者难以在模型的各个因素之间找到平衡，这些因素包含了：

* 简洁性（Simplicity）
* 规模（Size）
* 完备性（Comprehensibility）

要使用Petri网络模型来描述工作流，先应该对工作流中的基本模型部件有所了解。

### 工作流中的基本概念

###### 案例（Case）

工作流程（Workflow）的基本目的就是处理一个个的案例，例如绩效考核、保险索赔、贷款申请等等都是单独的案例。每一个案例都有自己唯一的标识。某个案例在出现和消失之间总是处于某个特定状态，这个状态包括了三元组：

（1）和案例相关的属性，指出特定条件下该案例是否被执行或者忽略。

（2）已经满足的办理条件，用来说明案例的进展。

（3）案例的具体内容，包括了文档、文件、档案或者数据库。

###### 任务（Task）

任务泛指一般的工作单元（Work Unit），而不是某一从具体的案例活动。具体的案例被称为活动。为了区分这一点，我们引入了工作项（Work Item）和活动（Activity）的概念。工作项是马上要被执行的实际工作，而活动指的则是该工作项的执行。

###### 过程（Process）

过程表示任务是否要被执行，以及什么时候执行，以什么顺序执行。我们可以将过程理解为案例的蓝图。我们用过程定义案例的生命周期，从开始执行到结束执行。

###### 路由（Route）

网络的路由决定了任务之间被执行的关系，以基于流程的程序设计方法为例，一般的路由包括了顺序、并发、选择和循环四种形式的路由。下文将阐述用Petri网络建模路由的基本模式。

###### 启动（Start）

每个工作项是由一个资源（Resource）来实施的，资源本身也是出了其他工作项以外的工作启动条件之一。触发工作项启动的形式包括：

（1）人力资源驱动，比如某个员工到达岗位，或者员工有时间开始某项工作等。

（2）外部事件驱动，例如收到了一个邮件或微信消息等。

（3）时间信号驱动，比如每周三下午2点开始一个任务等。

显然，触发是由外部环境而非工作流系统负责的，属于外部依赖，不在本文工作流建模的研究范畴内。

### 工作流到Petri网的映射

###### 工作流Petri网

用来表达工作流网络的Petri网络属于一种特殊的Petri网络。当一个Petri网络满足以下条件时，Petri网络是一个工作流网络（Workflow Net）：

1. 开始流程：N包含一个输入库所，使得，即没有前置库所。
2. 结束流程：N包含一个输出库所，使得，即没有后置库所。
3. 连接性：每个属于P的库所都位于一个从i到o的运行路径上。即没有多余步骤。

###### 过程建模

工作流的过程是由工作项的条件和具体任务组成。映射到Petri网的概念，库所就是条件，而变迁就是任务。条件和库所都是被动元素，而任务和变迁都是主动元素。案例就是令牌，案例的属性可以通过颜色扩展令牌来映射，令牌的值包含了案例的属性值。比如保险索赔案例的属性：赔额、索赔人、时间等等。

###### 路由的建模

可以通过Petri网络的结构，来表达各种工作流的路由，依次分析如下：

（1）顺序路由，对应图4-3，任务A和任务B是顺序执行的，任务B的输入是任务A的结果。通过在两个任务之间引入一个库所来解决，中间的圆圈对应的库所是任务B执行前的必须满足的条件，同时是任务A执行的结果。

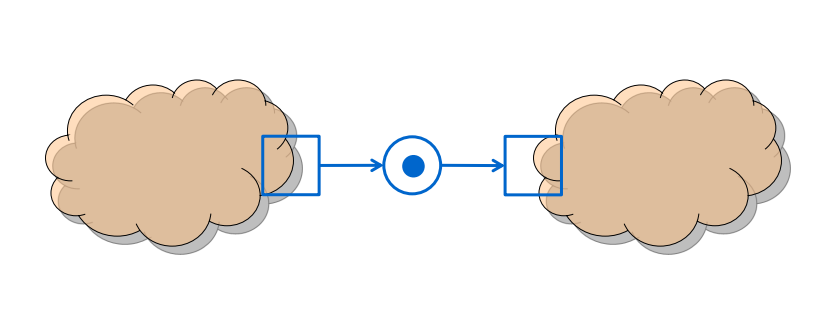


图 4-3 顺序路由Petri网

（2）并行路由：图4-4对应于一个互相无关的Petri网络：

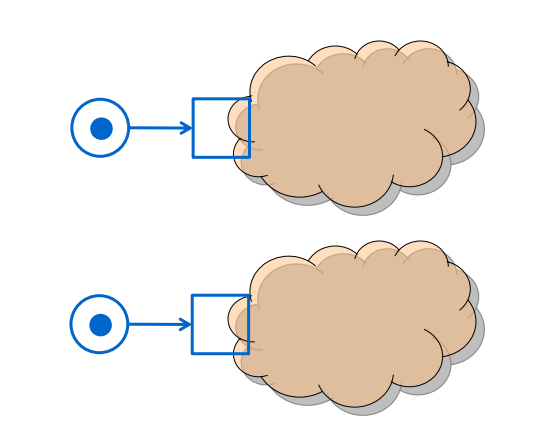


图 4-4 并行路由Petri网

我们更多地会遇到有相关的两个并行路由如图4-5：

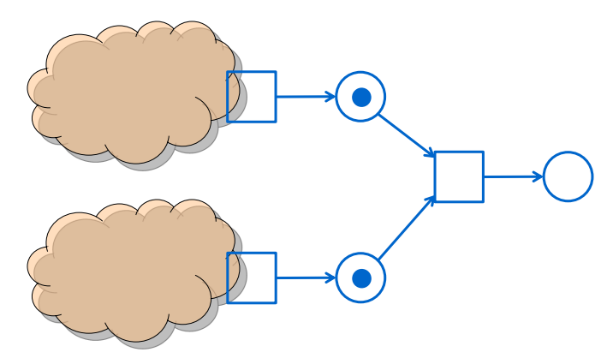
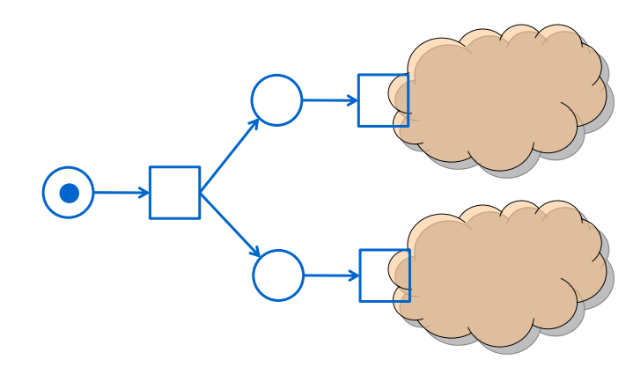


图 4-5 有关联的并行路由Petri网

为了并行地执行任务B和C，引入了任务A，称为And-Split，在A和B、C之间引入两个库所，当A任务实施后，为两个输出库所产生令牌，任务B和C就处于就绪状态可以激发。当B和C都实施之后，类似的实行And-Join任务（任务D）合并两个任务。

（3）选择路由：图4-6表示了选择路由的建模，选择执行B或者C。那么在B和C之前引入两个新任务t11、t12和两个库所，称为Or-Split。在前一个库所的令牌，或者实施t11，或者实施t12。假如我们实施t11，那么任务B将就绪，反之则任务C就绪。同样的可以建模Or-Join。选择路由还根据选择的时刻划分为两类，在此不再展开。

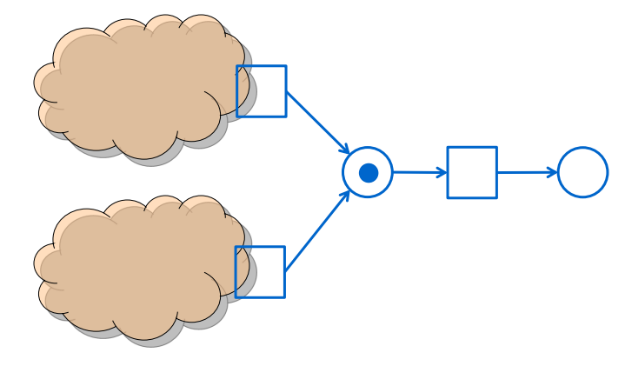
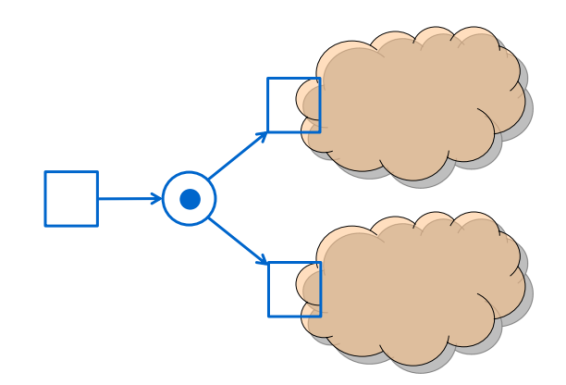
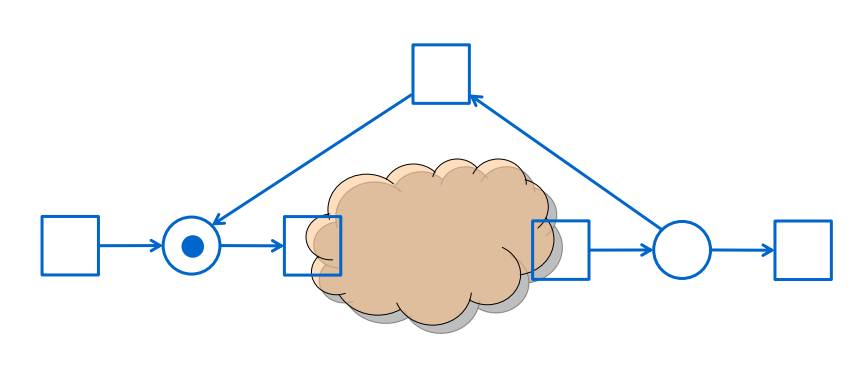


图 4-6 选择路由Petri网

（4）循环路由，和编程程语言中的循环类似，分为：Repeat ...Until...和While ...Do...两种，前者至少执行一次，而后者可能不执行.



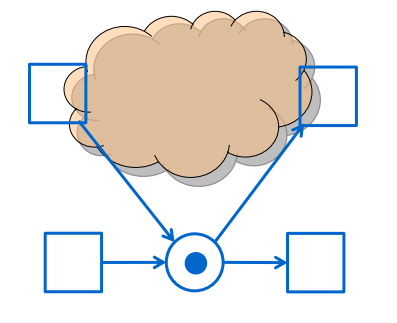


图 4-7 循环路由Petri网

（5）对库所容量限制的建模，或者同时处理某个任务的资源限制，可以用图4-7建模：

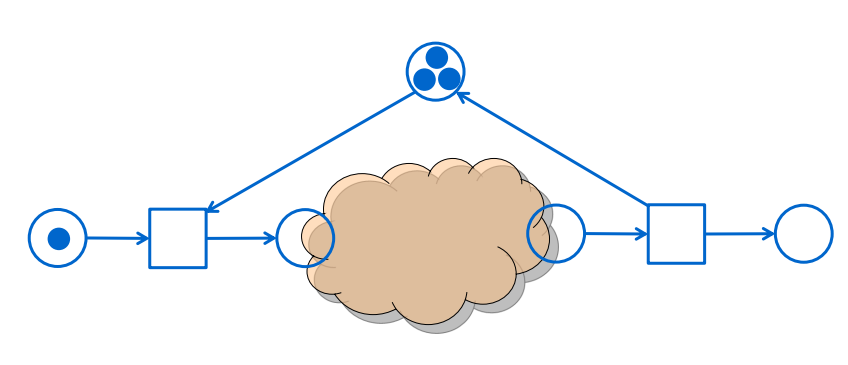


图 4-7 库所容量Petri网

（6）互斥和交替是常见的复合路由工作流，可以用图4-8和图4-9建模：

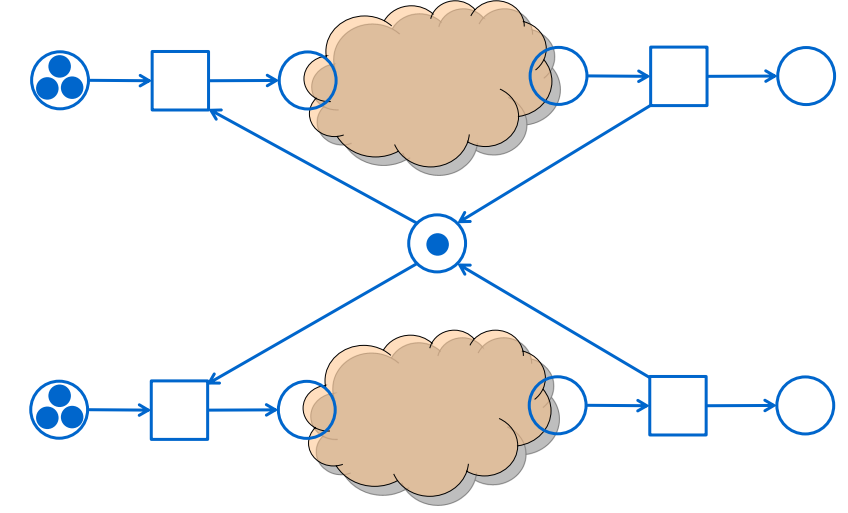


图 4-8 互斥路由Petri网

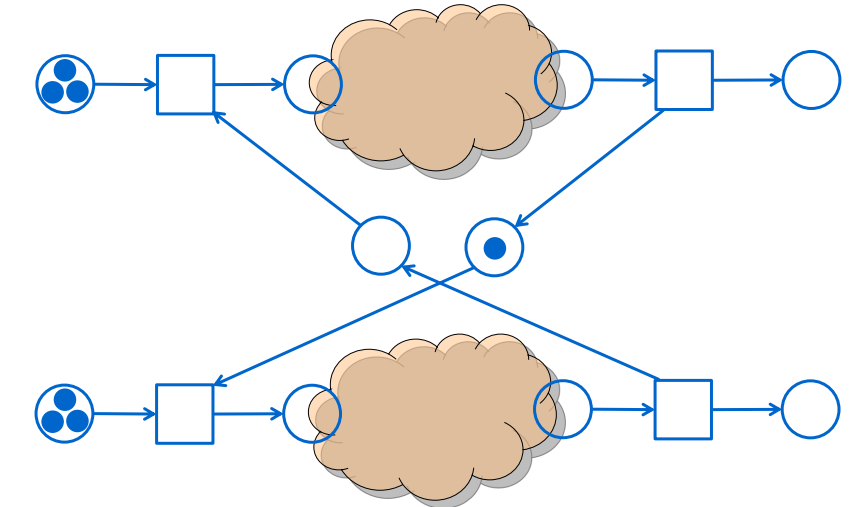


图 4-9 交替路由Petri网

###### 启动

我们看到，工作项就是将许多具体的案例加上即将要执行的任务的总和。一旦工作项被实际执行，它就变成了活动，所以活动是一个工作项的具体执行。映射到Petri网，工作项就是一个被激活的变迁（Enabled Transition），而活动对应于一个实际的变迁激发（Transition Firing）。

另外，有的建模使用的Petri网变迁是“饥饿”（Eager Firing）。也就是说一旦变迁激活了就会马上激发，这样的网络被称为自动激发网络。显然，我们的工作流中的激发不是自动的，而是“懒惰”（Lazy Firing）。它要求激发满足一些外部的条件，而这些条件没有在Petri网络中得到表示。就像前文所述，它有可能是人力资源驱动、时间信号驱动或者外部事件驱动的。为了在Petri网上表示这些激发，我们可以变迁符号的上面添加小图标来区分。通常用一个向下的粗箭头表示人力资源驱动，表示投入资源；而用时钟表示时间信号驱动，表示定时触发；用一个信封来表示外部事件驱动，表示收到邮件。

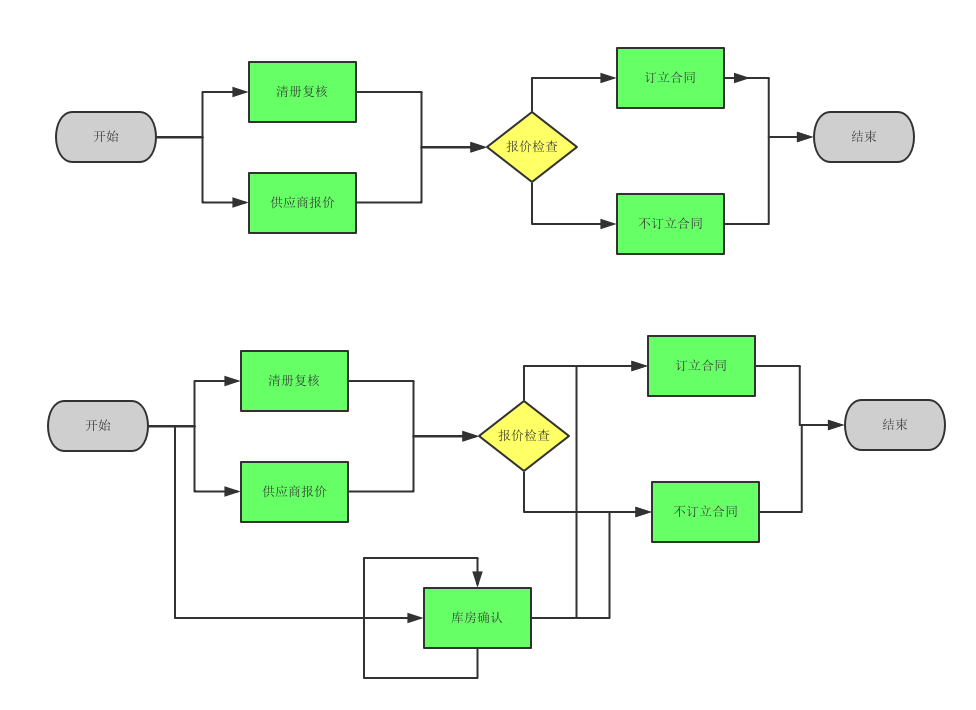
### Petri工作流网实践

上几节我们了解了Petri网，确实是对工作流甚至业务过程建模的良好工具，对于利用Petri网进行过程分析，还待进一步学习和实践。

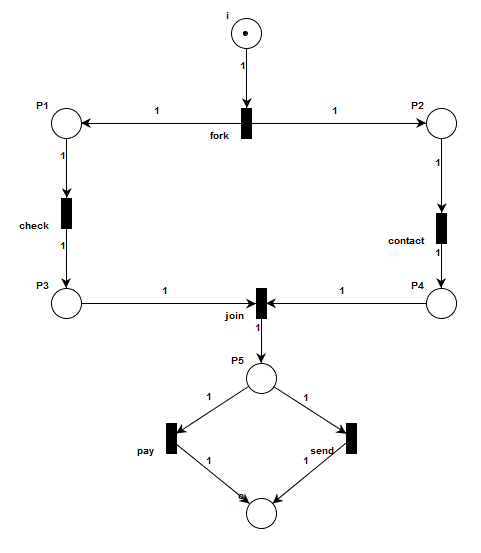
作者目前从事航空制造企业的金属材料采购工作，航空器生产过程中所涉及的金属材料范围非常广泛，除了常用的一些黑色金属以外，还会常用到一些有色金属，例如铝板、铝合金型材等等。铝合金材料是世界公认的最适合生产航空器的材料之一，目前对于飞机生产来说，金属材料是一种主要物料，一架飞机机身的大部分均由金属材料构成，可见金属材料是直接影响到飞机质量的物料，一旦金属材料的型号或性能改变，也将对飞机飞行过程中抗气流、抗压、抗寒等各种性能产生影响。因此在飞机的物料清单中，金属材料可以说是一种非常重要的物料，为了更好地把控它的质量，其采购途径及流程得到了比较高的关注。

目前作者所在企业的采购流程较为传统，随着大飞机科研生产的进度加快以及支线飞机批量生产的要求，一些生产目标不得不提前完成，而物料供应也必须跟上这样的需求。根据本文前几章节的分析可以得知，原本的采购流程存在很多缺陷，例如采购流程太过繁复、物料标准化程度低、采购缺乏计划性、采购策略单一等等，必须通过优化和再造流程使采购业务提升效能，因此作者根据日常采购工作的流程，并对这些流程进行了深入的思考。然而在以往的工作经验中，作者并无获悉在航空制造企业中采用Petri网络进行采购流程再造的案例，经过查阅和学习了诸多文献资料，作者也并未发现类似案例。基于Petri网络这一工具的属性，它既能够用图形来进行直观的流程描述，同时又具有数学属性，可以通过数学验证来分析流程的可行性，因此作者认为，对于航空制造企业中非常复杂的采购流程，Petri网络是一种较合适的工具，采用Petri网络对流程进行建模，作为采购流程再造的初步探讨和验证依据，这也是本文的创新点之一。

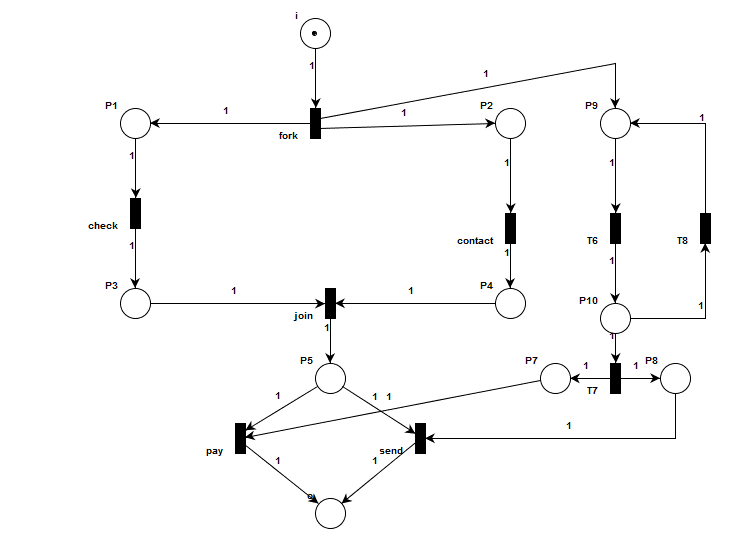
Petri网络对金属采购工作流建模和流程再造以后的建模如下：



可以建模为Petri网络



调整流程再造后的Petri网络调整为：



其意义为

表 4-？

|  |  |  |  |
| --- | --- | --- | --- |
| T0 | 登记 | P0 | 开始 |
| T1 | 清册复核 | P1 | 清册管理 |
| T2 | 供应商报价 | P2 | 供应商管理 |
| T3 | 报价检查 | P3 | 清册完成 |
| T4 | 订立合同 | P4 | 供应商完成 |
| T5 | 不订立合同 | P5 | 采购员询价 |
| T6 | 库房确认 | P6 | 库房管理 |
| T7 | 确认无货 | P7 | 库房进货 |
| T8 | 确认有货 | P8 | 通知有货 |
|  |  | P9 | 通知无货 |
|  |  | P10 | 结束 |

## 第四节 Petri工作流网络的扩展

针对上文提到的传统Petri工作流网具有一些缺点，比如数学表示容易变的太抽象、太庞杂而难以理解，又如无法对某些复杂活动进行有效的建模，再如无法有效地表达工作流的资源分配。这都对经典的Petri网络提出了挑战，目前研究领域提出了一些“高级Petri网”，其含义就是在各个方面进行了扩展。本文主要关注了扩展中的三种：时间扩展、层次扩展和随机扩展。

### 时间扩展

当需要对一个流程的性能进行量化分析时，可以为Petri网络引入了时间扩展。虽然Petri网络的理论基础——相对论否定了全局事件，但是在人类工作流这个维度，我们还是可以假设整个Petri网络是置于一个全局时钟之下，可以为每个令牌加上时间戳。只有当被消耗的令牌的时间戳早于当前时间，就绪的变迁才可能激发，这相当于引入了额外的一项隐含的激发前置条件，称为可用性（Availability）。产生的令牌的时间戳就等于最晚就绪的令牌时间戳+介于就绪和激发的等待时间+变迁发生延时。而对于前文所述饥饿Petri网络，规定了等待时间为0，即一旦所有的令牌都可用，变迁必须立即激发。

举个实际的例子引入时间扩展，我们可以对十字路口红绿灯的切换过程进行时间敏感建模，从而调整红灯、黄灯和绿灯的切换时间。变迁延时能够更精确地建模那些需要消耗事件的工作流人物，也帮助了我们理解整个工作流的一些性能指标，例如整体流程耗时和关键步骤的耗时，从而找到流程的瓶颈。

令牌上的时间戳是用特定的时间单位标识的，从分钟到纳秒不等，或者也可以引入工作流特定的时间单位，例如人月（man-month）或人日（man-day）。每个时间节点，网络都会检查每个变迁是否就绪。

另外，可以还可以定义有色的时间Petri网络和随机的时间Petri网络，例如执行某个任务的延时和操作人员的经验有关，且任务执行时间长短的确定性，也和经验相关，就可以在弧描述上定义和颜色属性有关的延时函数，某个令牌参与变迁的时间戳就将加入令牌的时间戳，从而影响变迁的可用性。

### 层次扩展

Petri网络是由一系列的库所、变迁、有向弧以及连接到其他元素的子过程组成的。为了反映这样的从上到下或者自下而上的层次结构，适应复杂的建模过程，可以为Petri网络引入层次扩展。对复杂网络的结构性能分析往往比较费时费力，通过将大网络分解为小网络，然后对大网络的结构分析。对小网络进行性能分析，业务流程再造优化才有意义。可以将大的Petri工作流网络分解为几个小的工作流网络，网络和网络之间通过库所连接，大网络中每个小网络都用一个开始库所和一个结束库所以及一个子网络变迁来表示，每个小网络内都从开始库所开始，结束库所结束。从大网络的角度看来，每个小网络代表了每个子流程的开始和结束。从大网络的角度观察流程，往往能够得到好很好的大局观，而从小网络的角度观察流程，往往能够集中某个相关流程的细节问题，因此结合了大网络和小网络的层次扩展，两者的分析之间相得益彰，不影响正确性的同时提高了效率。

### 随机扩展

Petri网络除了能够描述并发的流程以外，其对于不确定性的建模也是一大特色。传统的Petri网络就是一个不确定的网络，其激发与否、激发顺序、激发延时都是随机的。随机Petri网络就试图在概率论、数理统计和随机过程的基础上，对Petri网络的随机性之进行建模。

首先可以为Petri网添加优先级，用来表示即使这个变迁可以就绪了，也可以不被激发，应为有另一个就绪的变迁拥有更加高的优先级。因此，变迁的激发针对的是不同的优先等级，例如只有在优先级1和2都没有就绪的情况下，才能激发就绪的优先级为3的变迁。这实际上就为激发顺序建立了模型。当然实际处理的时候，会碰到所有优先级系统都需要处理的通用的问题，例如同优先级的排序问题，低优先级迟迟不能激发的饥饿问题，前者可以使用随机过程来建模，后者可以动态调整优先级。

Petri网络的非确定性属性研究一直是非常有价值的，因为它允许用户将大量的随机属性抽象成随机过程。其中最重要的两点，一个是变迁激发和激发延时，另一个则是变迁自身的执行延时。对于一个带时间的Petri网络，有的变迁有延时设定，有的没有，有的有确定的延时设定，有的有随机的延时设定，并有这些种类可能在运行的过程中发生转变。随机Petri网就作为一种时间Petri网的子类型应运而生，通过增加变迁演示的随机属性，例如偏移（dirft）和易变性（volatility），调整变迁的随机延时。通常工作流模型中常用的随机分布是指数分布和维纳分布。在这种情况下，Petri网的可达图是一个马尔可夫链。