一种面向企业用户的工作流模型

罗海滨　范玉顺　吴澄
清华大学 自动化系，北京 100084

摘要：工作流技术作为一种实现企业过程集成的有效手段，越来越受到人们的重视，并进行了广泛的研究。本文介绍了一种面向企业用户的工作流模型，从模型的角度描述了组成工作流的基本元素类型，并讨论了工作流模型与 Petri 网模型的转化，最后还给出了模型的面向对象的设计实现方案。

关键词：工作流　工作流管理系统　Petri 网　CIMS

1、前言

进入九十年代以后，伴随着企业信息化、网络化的逐步深入，企业内部的网络基础设施与工作组计算环境日益完善，这就为工作流技术在企业中的应用、实施提供了可能的条件。作为一项集成技术，工作流所要实现的目标就是使企业中大量的基于知识与规则的任务和活动能够相互协调一致、高效运作，在正确的时间能够将正确的信息传递给正确的人从而完成正确的业务流程。从这一角度来看，工作流技术并非是简单的一个计算机应用，而是企业信息、企业资源、企业应用与企业人员的集合体。

根据工作流管理联盟（WfMC）所提出的工作流系统体系结构[1]，一个完整的工作流管理系统具有包括建模工具、工作流机（也称工作流服务器）、管理工具以及用户界面在内的分布式结构。从这一体系结构来看，工作流模型是整个系统的基础，它的确定性保证了系统内各组成部分之间交互的一致性，也决定了一个工作流系统从设计开发到运行实施的诸多环节，不同的工作流模型也就形成了不同的工作流系统。

本文将介绍一种简单、直观，又具有较强描述能力的面向企业用户的工作流模型，详细介绍了组成模型的各类元素，并给出了与之相对应的 Petri 网模型，以便对其性能进行分析与评价，最后还给出了模型的面向对象的设计实现方案，以供开发人员参考。

2、面向企业用户的工作流模型

企业中一个完整的业务流程是由一系列最基本的生产活动按照一定的逻辑顺序规则组成的，这些生产活动与它们之间的逻辑关系可以很直观地映射成为一个由节点和连接弧组成的有向图。有向图中的节点即表示一个可执行的活动单元，连接于两个节点之间的有向弧即表示活动间的先后顺序关系。我们有如下定义：

（1）有向图 $G=\{N, L\}$为一个二元组，$N=\{n_1,n_2,\ldots,n_s\}$为节点的集合，$L=\{l_1,l_2,\ldots,l_r\}$为连接弧的集合，其中 $l_i=(n_j,n_k)$为从 n_j 指向 n_k 的连接弧，$nj,nk\in N$.

（2）对于$\forall\ n_i,n_j\in N$，若有 $l=(n_i,n_j)\in L$，则 n_i 为 n_j 的一个前趋节点，n_j 为 n_i 的一个后继节点，l 称为 n_j 的一条输入连接弧，或者 n_i 的一条输出连接弧。

（3）若 $N'\subseteq N$，且 $N'=\{n' \mid (n',n)\in L\}$，则 N′为节点 n 的前趋节点集，记为 Pre(n)；

若 $N'\subseteq N$，且 $N'=\{n' \mid (n,n')\in L\}$，则 N′为节点 n 的后继节点集，记为 Post(n).

（4）节点状态：对于 $\forall\ n\in N$，有状态函数 State(n)={0,1}，当节点 n 处于非执行状态时，State(n)=0；当节点 n 处于执行状态时，State(n)=1。初始时刻，$\forall n\in N$，State(n)=0。

（5）转移函数：对于 $\forall\ l\in L$，有转移函数 Trans(l)={0,1}，如果 Trans(l)=1，则连接弧 l 允许发生转移（是否发生转移，则要取决于后面的演进规则）；如果 Trans(l)=0，则连接弧 l 不能转移。

（6）演进规则：有向图的演进是由节点状态的改变与连接弧发生转移这两个动态因素

相互作用而完成的，因此，规则包含如下两个方面：

a、 对于$\forall n \in N$，

当 State(n)=0 时，若$\exists l=(n',n)$发生转移，则 State(n)=1；

当 State(n)=1 时，若节点 n 执行完毕，则 State(n)=0；

b、 对于$\forall l=(n,n') \in L$，

当 State(n)从 1 变为 0 时，若 Trans(l)=1，则连接弧 l 发生转移；若 Trans(l)=0，则连接弧 l 不发生转移，直至下一次 State(n)从 1 变为 0 的时候再使用此规则。

2.1 节点的类别

有向图中的节点代表了具有如下特征的多种实体：

（1）与企业中实际存在的事件或活动有着直接的对应关系；

（2）本身有着具体的或人为定义的含义；

（3）能与其他节点形成一定的逻辑关系；

因此，区分不同类别的节点、对节点进行具体的类别定义不仅可以明确节点的含义，同时也增强了模型的语义。在这里，我们赋予节点以如下的几种类型定义：活动、子过程、开始与结束标记、同步节点。

2.1.1 活动

活动是指在一段不间断的时间间隔内为实现某一目标由人工或自动完成的一个企业行为，是组成业务流程的最基本单元。一个企业的所有活动的集合覆盖了企业中各类业务流程的全部细节。虽然企业中的活动多种多样、千差万别，但是却可以用一个统一的结构化框架来描述它，如右图所示：

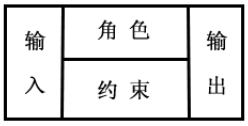


图 1 活动的描述框架

（1）输入：活动的输入部分是保证活动开始的物质条件，通常包括企业资源与信息对象。企业资源主要是指原始物料、中间产品以及生产设备等，而信息对象则表现为各类电子化文档，包括数据表格、图形文档、文本文档、电子邮件及 Web 资源等。

（2）输出：活动的输出部分是活动的结果，类似地，也包括两类典型输出——企业资源与信息对象。活动的输入与输出构成了每一个基本活动单元与外部（其他活动单元）之间的接口，封装了内部具体的任务处理过程，包括角色与约束；而活动输入与输出在内容上的一致也为实现工作流模型的重用提供了保证，通过建立相应的输出→输入映射机制，一个活动单元便可以同多个活动单元进行组合，出现在不同的业务流程当中。

（3）角色：角色是指企业中以一定的技能要求为前提、能够完成某项专职工作的企业人员的集合，它与企业的组织模型紧密相关。活动所需的角色包括执行者与负责人两类，二者在活动中形成了上、下级的关系，下级负责活动的执行，上级则负责监督、检查与异常情况的处理。

（4）约束：活动在执行的过程中总是有一定的约束条件，这也体现了竞争激烈的市场与独立自主的客户对企业所施加的压力。最主要的一个约束就是对活动的时间要求，即活动应该在有限的时间内必须完成。第二个约束是活动的优先级，赋予活动不同的优先级将区别了不同活动对企业的重要程度。优先级越高的活动在工作流实施运转的过程中将享有更为优先的申请企业资源与人员的权利。除了时间和优先级以外，根据企业本身的特点，还将会有许多其他方面的约束。

在统一的活动描述框架下，我们可以定义多种不同类型的活动。一方面可以方便用户建模，使用户更直观地理解具体活动的含义；另一方面，工作流机针对不同类型的活动，在实施过程中可以做出不同的处理，使系统更灵活、更高效。活动分类的标准有很多，可

以根据企业的实际情况来确定。比较基本的一种分类是将活动类型确定为人工型与自动型两种。人工型的活动是通过工作表的生成来通知相关的人员，依靠人员以手工或启动应用的方式来完成；自动型活动则是在工作流机的驱动下直接启动应用或利用自动化设备来完成的活动，例如工作流机自动启动某台计算机上的绘图应用程序并打印一份图纸。这种自动型活动充分体现了工作流管理系统所实现的企业内部不同应用间的过程集成。

2.1.2 子过程

作为组成业务流程的最基本单元，活动是不能被进一步分解的。一旦我们的流程比较复杂，涉及的环节比较多，那么活动的数量也将大大增加，有向图中的节点数也必然会不断膨胀。这首先影响了用户对流程中各主要环节的把握和理解。事实上，我们可以把某些关系紧密的活动集合起来，在图上以一个节点表示，这就形成了子过程的概念。

子过程是一类能够分解的节点类型，它的内部可以包含组成工作流模型的所有元素类型，实质上就是一个子工作流。子过程的引入大大增强了模型的表达能力，使模型具有了层次化的概念，并支持自顶向下的建模过程。我们规定，子过程可以出现在模型的任何层次，即允许子过程内部再次嵌入子过程。通常，用户可以在模型的最顶层全部用子过程来表示，这样即确定了模型的总体逻辑结构，进而再在每个子过程中详细地布置活动及其他模型元素，直到完成最底层的基本活动的建模。

2.1.3 开始与结束标记

由于有向图本身是一种非线性的数据模型结构，与线性结构不同的是，它可能具有多个入口节点（即只有连接弧由该节点发出，而没有连接弧指向该节点），这就给用户与工作流机正确理解流程的逻辑顺序带来了不便，甚至会发生疏漏与错误。因此，我们又人为地定义了一类具有特定含义的标志性节点——开始标记 B。我们规定，开始标记为一个工作流模型（或子过程）的唯一入口点，它无前趋节点，即 Pre(B)=Φ。

对于一个实际的业务流程，可能会由于不同的执行情况而出现不同的结果。对应于有向图，这种情况就表现为一次不能遍历图中的全部节点，只有部分路径被选择执行，图中会出现多个出口节点，它们标志着流程的结束。为了清晰地表达流程的结束状态，并与开始标记相对应，我们引入了另外一类标志性节点——结束标记 E，同时规定，结束标记 E 为一个工作流模型（或子过程）的唯一出口点，它无后继节点，即 Post(E)=Φ。

2.1.4 同步节点

在我们把一个实际的业务流程映射成为工作流模型时，很重要的一点，就是要保证活动间的逻辑关系不变。“与”和“或”是两类最基本的逻辑关系，它是表达各种复杂关系的基础，工作流模型必须具备表达“与”和“或”关系的能力。

我们在前面已经定义了模型的演进规则，对于任意一个处于非执行状态的节点 n，只要有一条输入连接弧发生了转移，那么该节点即可被执行，这实际上就表达了“或”的关系，即 $\cup\{n' \in Pre(n),\ l=(n',n) \in L,\ Trans(l)=1$ 且 l 发生转移$\}$。

对于“与”的关系，我们通过增加一类新的节点——同步节点 S 来表达，它对活动起协调、同步的作用。我们规定，同步节点 S 的动态行为完全遵循演进规则，所不同的是，当 S 处于执行状态时（State(S)=1），将判断它的所有输入连接弧是否已经全部发生转移：若是，则 S 的状态就由 1 变为 0，即 S 执行完毕；否则，S 仍处于执行状态，并将继续判断，直至满足上面的条件后 S 才执行完毕，State(S)=1→0。这意味着同步节点将使它的所有前趋节点都执行完毕后才继续推进流程，表达了“与”的关系，即 $\cap\{n' \in Pre(n),\ l=(n',n) \in L,\ Trans(l)=1$ 且 l 发生转移$\}$。

同步节点的设置不仅区分了有向图中节点的多条输入连接弧之间的“与”、“或”关系，而且可以使用户从图中清楚、直观地理解流程的执行过程，既丰富了模型的语义，也方便了用户建模。

2.2 连接弧

作为有向图中的另一类组成元素，连接弧表达了图中不同节点元素之间的逻辑顺序关系。它从前趋节点指向后继节点，体现了节点状态的转移与有向图的演进。

连接弧发生转移是有条件的，因此我们在前面的定义中为每一条连接弧都绑定了一个二值的布尔型转移函数 Trans(l) 。对于转移函数的组成，我们有如下定义：

转移函数是由一系列条件（Conditions）经过“与”、“或”组合而成的，其中每一个条件就是一个谓词逻辑，它的结果也是“真”、“假”二值的，这些逻辑的最终组合结果决定了转移函数的取值。

根据连接弧转移条件的特点，我们把在工作流模型中所应具体表达的连接弧类型分为如下两类：

（1）永真型：即连接弧的转移函数值永远为“真”，Trans(l)≡1。这体现了一种顺序关系，不需要经过任何条件的判断，只要前趋节点执行完毕，即可激活后继节点。

（2）不定型：即转移函数的取值是需要在具体的工作流执行过程当中由工作流机或人加以判断来确定的。这种判断实际上体现了一种选择关系，即根据不同的情况，通过满足条件的连接弧的转移，实现对某一节点的多个后继节点的选择性激活。

从直观和方便的角度出发，我们把“永真型”的连接弧称为“无条件连接弧”，而把“不定型”的连接弧称为“有条件连接弧”。这两种连接弧在描述活动节点间的时序关系与逻辑顺序的同时，也隐含了交换数据的内容，即前趋节点的输出可以做为后继节点的输入。但这种表达方式处理数据流的能力是很有限的，比如，对于那些不具备前趋/后继关系的节点就不能显式地定义。为此，我们引入另外一种连接弧——“数据连接弧”。这种连接弧可以连接于同一层次中的任意两个有数据交换的节点之间来定义数据流，数据流动的方向就是连接弧箭头所指的方向。

2.3 一个工作流模型的实例

在阐明了组成工作流模型的各类元素之后，我们给出一个完整的模型实例。这个例子是描述某种产品的生产加工过程，如下图所示。

图 2 一个工作流模型的实例

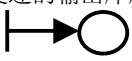
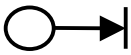
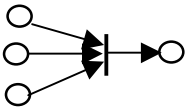
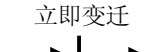
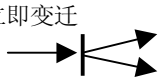
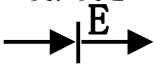
流程开始以后，由开始标记经过一条无条件连接弧的转移激活第一个“准备生产信息”的活动，在完成该活动以后，将经过两条无条件连接弧同时激活两个子过程 1、2，这两个子过程是并行的生产过程，其内部包含了一系列串行的生产步骤（图中只显示了最顶层的模型结构，未画出子过程内部的细节）。同步节点表示只有当两个子过程 1、2 全部完成以后，才进一步推进流程。后面的“领导决策”活动将对产品进行检验并根据结果选择执行子过程 3 或 4。条件 1 与条件 2 分别是两条连接弧上的转移条件，它们的取值将决定激活哪一个子过程。当子过程 3 或 4 执行完毕后，结束标记被激活，整个工作流也随之结束。

3、工作流模型向 Petri 网模型的转化

Petri 网模型[2]是目前系统建模领域中被人们广泛采用的一种方法，它在描述离散事件动态系统方面的能力已经得到公认，具有较强的模型分析能力。我们通过工作流模型向 Petri 网模型的转化，就能够借助较为成熟的 Petri 网工具，完成对工作流模型的评价。

我们按照下表将工作流模型中的组成元素转化成为 Petri 网模型中的组成元素（数据连接弧并未考虑在内，因为它并不构成过程的控制逻辑）：

表 1　工作流模型与 Petri 网模型对照表

工作流模型		Petri 网模型	解释说明
节点	活动	库所 	库所中的 Token 是有颜色的，分别代表活动的状态以及活动执行所需的资源与人员，有 Token 的库所表示活动正处于被执行（激活）状态，即资源与人员正处于被锁定状态。Token 的移出表示资源与人员的释放。库所是有时限的，并且内部包含了一系列的操作，这些操作的结果将决定库所的输出变迁是否能被激活。
	子过程	子网库所	可以用子网中的结构等价地替换该库所。
	开始标记	无输入库所的变迁+该变迁的输出库所 	该变迁是受外界因素控制的（包括人为因素），它自动产生 Token 以激活整个 Petri 网。
	结束标记	无输出库所的变迁+该变迁的输入库所 	该变迁将 Petri 网中的 Token 移走，以使整个 Petri 网处于非激活状态。
	同步节点	具有多输入库所、单输出库所的变迁+相应库所 	变迁的每一个输入库所分别连接同步节点的每一条输入连接弧。
连接弧	由某一节点发出的唯一的“无条件连接弧”	立即变迁 	表示顺序关系。
	由某一节点发出的多条“无条件连接弧”	单输入、多输出的立即变迁 	表示并发关系。
	由某一节点发出的每一条“有条件连接弧”	受控变迁 	表示选择关系，只有满足受控条件的变迁才能发生转移。

参考以上的对应关系，可以把任何一个工作流模型转化为 Petri 网模型。如图 3 所示。

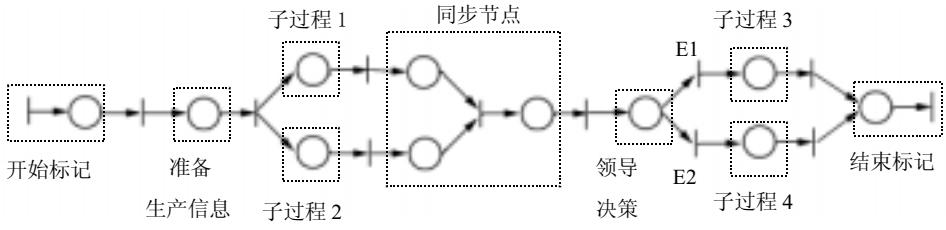


图 3　工作流模型向 Petri 网模型的转化

我们把图 2 给出的工作流模型转化成了 Petri 网模型，并在图中用虚线框标明与之相对

应的图 2 中的节点，E1 和 E2 表示的是条件 1 和条件 2。

4、面向对象的模型实现方案

基于前文对工作流模型的详细阐述，本节讨论模型的实现方案问题。在模型的设计实现过程中，有以下几个问题需要考虑：

（1）模型中的活动类型最好能够根据企业的实际应用情况进行不同的扩充，或者能够提供给用户以充足的活动模板支持，使用户可以方便地定制自己的企业活动；

（2）在用户建模、浏览时提供图形用户界面（GUI）是必须的，因而需要解决模型中各组成元素的可视化问题；同时，模型在被用户使用时需要记录大量的数据，那么应该如何保存这些数据也是需要解决的一个问题；

（3）模型中有些组成元素的行为表现是基本相同的，仅存在着一些细微的差别，这一特点应该被充分利用，以提高开发效率；

（4）模型中各组成元素并不是孤立的，相互之间是有联系的，是一个统一的整体，采用什么样的机制来组织、协调模型中的各个元素将直接决定模型实现方案的合理性。

兴起于九十年代的面向对象技术为软件的开发提供了一种新的模型，它具有传统的面向过程方法所不可比拟的优点，种种支持面向对象的 CASE 工具也使开发者与最终用户受益非浅。针对工作流模型的有向图结构，我们对面向对象的设计、开发方案做如下分析：

（1）工作流模型中的每一种组成元素都对应于一个类的实现，利用对象的封装性把属性和方法（成员函数）结合在一起，将简化程序的设计和改进，而且使模型元素的扩充变得易于实现，开发人员只需要再相应地增加一个新的类即可；

（2）类中的属性对应模型元素所要记录的用户数据和可视化的状态（比如该元素是否处于被选中状态等），类中的方法（成员函数）则提供了模型元素的各种操作并解决了元素的可视化问题；

（3）工作流模型中的元素可以分为节点和连接弧两大类，利用面向对象的继承性，我们可以设计两个基类，使不同的节点类型和连接弧类型分别从这两个基类中派生，以实现公共部分的代码复用；同时，利用多态性在不同的派生类中实现基类中同一成员函数的不同的响应细节，这将有利于减少程序的代码量；

（4）面向对象的消息传递和响应机制为协调和组织模型中的各个元素提供了解决的方法，消息成为对象间相互作用的唯一通道，这使得对象间的边界清晰、通信显明，有利于实现细节的隐蔽与程序的模块化设计。

图 4 给出了一组工作流模型组成元素的类的设计图，图中的注释和标记采用 Booch 方法的标准[3]。

在图中，CUnit 和 CLink 分别对应节点类和连接弧类的两个基类，其中 CUnit 类是一个抽象基类，CLink 类则对应于无条件连接弧；由 CUnit 类派生出了六个子类，分别对应人工型活动（CManualTask 类）、自动型活动（CAutoTask 类）、子过程（CProcess 类）、开始标记（CStart 类）、结束标记（CEnd 类）和同步节点（CSyncNode 类）；由 CLink 类派生了一个子类，对应于有条件连接弧（CLinkCon 类）；图中 CProcess 类包含了除抽象基类 CUnit 以外的各个类，对应于工作流模型中的子过程可以包含组成工作流模型的各类元素（包括子过程本身）；CText 类对应于显示在模型图中的注释。

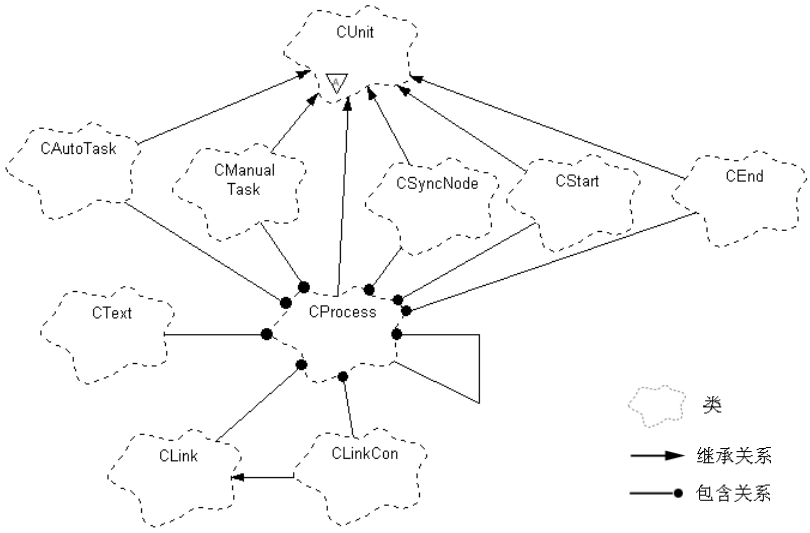


图 4 工作流模型的设计类图

5、结论

本文着重讨论了一种面向企业用户的工作流模型，给出了完整的定义，并描述了相应的 Petri 网模型转化方法，最后对面向对象的模型实现方案进行了分析。这种工作流模型具有明显的面向企业用户的特点，简明、直观，适于描述企业内部的各类经营过程，适于集成企业内部的各类应用，对经营过程重组能够提供一定程度的支持。在本文所提出的工作流模型的基础上，我们已经进行了相应的工作流建模工具的开发。

工作流建模技术与企业经营过程建模（BPM）是有着密切关系的，其中所涉及的形式化描述语言、模型的分析与优化以及模型的仿真等方面的内容，是将要进一步研究的问题。

参考文献：

[1] Workflow Management Coalition. The Workflow Reference Model, TC00-1003 [Z]. Hampshire, UK: Workflow Management Coalition, 1995.

[2] Peterson James L. Petri Net Theory and Modelling of Systems [M]. New Jersey: Prentic-hall, Engle-wood Ciffs, 1981.

[3] Booch G. Object-Oriented Analysis and Design with Applications, Second edition [M]. New York: Addison-Wesley, 1994.

An Enterprise User Oriented Workflow Model

Luo Haibin　　Fan Yushun　　Wu Cheng

Department of Automation, Tsinghua University, Beijing 100084

Abstract: Workflow technology is an effective way to implement process integration in enterprises. It is focused much attention and studied widely in recent years. In this paper, a workflow model which orients enterprise user is introduced and the fundamental elements of the model are discussed. Also a method which converts the workflow model to the petri net model is given. At last, an object-oriented design scheme is presented to the developers.

Key words: Workflow　　Workflow Management System　　Petri Net　　CIMS