

# 工作流过程建模方法及模型的形式化验证<sup>\*</sup>

杨东 王英林 张申生 傅谦

(上海交通大学计算机系 CIT 实验室 上海200030)

## A Survey of Workflow Modeling Approaches and Model Verification

YANG Dong WANG Ying-Lin ZHANG Shen-Sheng FU Qian

(Department of Computer, Shanghai Jiaotong University, Shanghai, 200030)

**Abstract** Workflow technology is widely used in business process modeling, software process modeling as well as enterprise information integration. At present, there exist a variety of workflow modeling approaches, which differ in the easiness of modeling, expressiveness and formalism. In this paper, the modeling approaches most used in research project and workflow products are compared. And the verification of workflow model is also dealt. We argue that a ideal workflow modeling approach is a hybrid one, i. e. the integration of the above approaches.

**Keywords** Workflow modeling, Model verification

## 0 引言

工作流技术最初起源于办公自动化和企业生产中的过程概念。包括办公室内的文档路由和生产中活动的路由。目前,还没有统一认可的工作流定义。工作流管理联盟(WfMC, Workflow Management Coalition)给工作流的定义是“商务过程的全部、部分自动化或计算机化”<sup>[2]</sup>。一个完整的工作流系统,也称工作流管理系统,应包括两部分:(1)过程建模部分(工作流建模)。(2)工作流引擎部分,为模型的运作提供环境。其中,工作流建模部分在工作流技术中占有重要的部分。工作流模型的精确定义与否影响着工作流引擎的性能,从而决定着企业的关键商务过程的运行效率,也决定了企业的商务目标的成功与否。一个好的工作流建模既要为普通用户所掌握,还要能为计算机所解释。目前主要的工作流建模方法有:

- 基于程序语言的方法
- 基于图的方法
- 基于逻辑的方法
- 基于规则的方法
- 基于代数的方法

## 1 基于程序语言的方法

程序设计语言可以用来描述工作流中的控制流和数据流。例如,在 MOBILE 中,控制流用由一些特定的控制流元素序列组成的控制流过程来定义<sup>[36]</sup>。ProcessWEAVER 提供了一种可被解释的脚本语言,它包括一些内在函数以处理表的消息传递,处理数据访问和程序调用,并且能够被扩展以提供扩展的功能。另外一种工作流过程语言是 APPL/A<sup>[35]</sup>,它对 Ada 语言进行了扩展,以适合描述(软件)过程规范。这种方法的主要缺点是缺少对工作流规范进行形式化分析和验证的理论基础。这就意味着工作流规范的正确性仅仅能够以执行来保证。此外,由于并没有提供对组织视图的支持,需要另外的方法来定义。

## 2 基于图的方法

目前,大多数工作流建模采用了基于图的方法,因为基于

图的方法通常具有图形可视化的优点。它们大都提供了类似状态迁移图的图示化方法。基于图的方法主要包括:基于活动网络图的方法、基于 Petri 网的方法、基于 Statechart 的方法。

### 2.1 基于活动网络图的方法

IBM 采用了基于活动网络图的方法,工作流过程用一个有向图表示<sup>[13]</sup>。图中的节点表示活动或任务,节点间的连接弧表示工作流中的控制流和数据流。组成工作流模型的元素包括过程、活动、模块、控制连接弧、数据连接弧和条件。其中,模块类似于“过程”,但其不同于过程之处在于模块没有唯一的标识符,只能被一个工作流过程使用,模块也有输入、输出数据箱。控制连接弧定义了两个活动之间的执行顺序。当连接弧的起始节点执行完毕后,工作流引擎将会根据控制连接弧的定义进行过程导航。控制连接弧与它们所连接的活动节点构成了整个工作流过程的控制逻辑。而数据连接弧定义了活动间的信息流,由前一个活动输出数据箱指向后一个活动的输入数据箱,数据箱中包含了工作流相关数据和与活动执行相关的具体应用数据。条件用来定义工作流执行中的约束,主要包括属于活动外部的转移条件和属于活动内部的开始与结束条件。转移条件使定义在控制连接弧上的条件,根据对转移条件的判断结果是“TRUE”或“FALSE”来决定是否执行下一个活动。开始条件定义了活动在什么情况下才能开始执行。设置开始条件和转移条件从而区别了活动的使能和活动的被执行。结束条件定义了活动在什么情况下才能够结束。通过结束条件的设置,可以定义需要多次循环执行的活动。但文[13]中并没有形式化定义活动网络图的语义,因而难以对模型进行验证。

### 2.2 基于 Petri 网的方法

Petri 网具有图形化的界面,并具有良好的数学语义,因而成为工作流建模的主要方法之一,目前国际上对将 Petri 网运用于工作流的建模之中进行了大量的研究,也出现了不少的研究成果。文[3]用 Petri 网的基本元素—库所(Place)和迁移(Transition)定义了 WfMC 规定的六个基本元素,记 AND-split, AND-join, OR-split, OR-join, 循环以及顺序。它采用库所来建模活动(任务),迁移来表示控制活动。而工作流过程则采用具有颜色和层次的高级 Petri 网来表示。最后,得到

<sup>\*</sup> 863/CIMS(2001AA412010)。

的工作流模型能够用工具 NCUPN 进行分析。文[4]采用了集控 Petri 网来建模工作流过程。集控 Petri 网是一种扩展的有色 Petri 网,它在有色 Petri 网的基础上,增加了过程变迁、不确定弧、全局库所等。集控 Petri 网中的托肯分为普通托肯和有色托肯。普通托肯对应着活动的功能输入/输出对象,有色托肯对应着支持活动的资源,包括人力资源和物力资源。INFCOM 采用了嵌套关系/迁移网(NR/T 网)来建模工作流<sup>[5]</sup>,NR/T 网是一种新型的高级 Petri 网,它集成了并发进程建模和复杂结构化对象(也称嵌套关系)。在 NR/T 网中库所对应于一种复杂结构化对象,迁移表示对输入库所和输出库所的操作。而库所对应的复杂结构化对象用语义数据模型描述(如 ER 模型)。ELLIS 采用了信息控制网 ICN 描述工作流模型。信息控制网是一种在 Petri 网-E 网的基础上,扩展了数据流的表示。它由活动节点(库所)、OR 控制节点(迁移)、AND 控制节点(迁移)组成。它还有仓库节点(Repository),以表示持久数据。此外,信息控制网的形式化表示上还增加了完成活动的角色和人员的映射信息,并引入了 SendTo 原语以进一步表示工作流的例外处理<sup>[6]</sup>。Van der Alast 采用了 WF-net 网表示工作流模型<sup>[6]</sup>,其中,活动和控制元素都用迁移来表示,两者之间采用辅助的库所来连接。

Nabil 采用了不同的方法来表示任务,它建模了细粒度的任务状态。在它的 PN 框架中,任务不再用单一的库所或变迁来表示。与此对照,任务被建模成有一组表示任务状态的库所和表示任务间状态转移的 Petri 网迁移。任务的状态被区分为:INITIAL, EXECUTION, DONE, COMMIT, ABORT<sup>[10]</sup>。文[7]采用了“任务结构”来表示任务。任务结构是由一组 Petri 网的库所和迁移组成,这些迁移表示了任务在请求资源、分配资源、执行活动、完成活动以及释放资源等方面的生命周期过程。因而,在这种方法中,任务的 Petri 网表示包含了引擎的调度资源的信息。文[11]采用了 Task-Net 来建模工作流中的任务。同样,任务用包含一组 Petri 网的库所和迁移的任务结构表示。任务结构表示了支持事务的任务状态以及状态间的转移。任务的状态包括 INACTIVE, RECEIVED, WORKING, SUSPEND, COMPLETED, COMPENSATED 以及 FAILED, ABORTED。

从上面可以看出,采用 Petri 网对工作流进行建模是一种常用的方法。其基本思想是采用、或扩展常规的 Petri 网,以表达工作流中活动间的关系。但归纳起来,上述各种建模方法大致可以分成以下三类:

- 1) 用 Petri 网的库所表示任务,变迁表示控制元素(如文[3,6])。
- 2) 用 Petri 网的变迁表示任务,库所表示控制元素(如文[5]等)。
- 3) 用一组变迁和库所来表示任务。通常,这组建模了细粒度的任务。例如,文[11]中的任务结构表示了支持事务的任务状态间的转移。而文[7]也称任务结构,但它却建模了工作流引擎对活动的资源调度关系,将资源的分配、调度纳入工作流模型中。文[10]也采用了类似的细粒度任务表示。

前两种方法建模了粗粒度的任务,即任务用单一的 Petri 网元素表示。而后一种建模了细粒度的任务,即任务描述成一组变迁和库所的集合。当用单一的元素来表示任务时,方法一和方法二存在区别。尽管 vander Alast 认为方法二能够区别任务的使能和任务的执行<sup>[9]</sup>。但方法二的方法造成工作流模型中的辅助库所太多,使图形过于复杂。相对来说,方法一用单一的库所来表示任务,它说明系统处于任务执行的一个稳定期间内,这有点类似于 Statechart 中对活动的表示,其优

点是图形元素少,因而工作流模型十分紧凑。这从文[6]和[9]的工作流模型的对比中可以看出。因而,采用方法一的工作流建模系统占有更大的比例。反之,如果需要建模细粒度的工作流系统时,则采用方法三。将任务描述成一组变迁和库所的集合。根据建模需求的不同,这些变迁可能表示资源的分配,或者表示任务的各个状态、或者具有事务性质的任务。Petri 网具有严密定义的语义,因而能够对模型进行形式化验证。

### 2.3 基于 Statechart 的方法

Statechart 是由 David Harel 提出的,它是对有限状态机的扩展,是一种描述、分析复杂反应型系统的图示化手段<sup>[15,17]</sup>。目前,它已经广泛地用于描述实时系统、软件过程的动态行为。它表达力强,容易理解,具有形式化语义等特点。

Statechart 本质上是对有限状态机的扩展,它主要由状态和状态间的迁移组成。状态间的迁移可带有标记,通常以 ECA(事件-条件-规则)规则的形式,即 E[c]/A 的形式出现。Statechart 主要特点是提供了深度分解和正交分解,从而提供了用户在不同的抽象层次上描述系统,以简化或进一步细化系统,并有效地减少了状态数目,避免了有限状态机等存在状态爆炸的问题。

Muth 在 MENTOR 项目中首次使用 STEAEMATE 来进行工作流建模<sup>[12]</sup>。它包括 Activity-chart 图和 Statechart 图。其中,Activity-chart 图反映了系统的功能分解,相当于功能视图的作用。Activity-chart 图的节点表示活动,有向边描述了活动之间的数据流关系,有向边上标注了活动之间流动的数据项。而 Statechart 图用来表示系统的动态行为,它描述了活动间的控制流关系。为了实现分布式的工作流运行,在 MENTOR 项目中,对 Statechart 图和 Activity-chart 图进行了分割,位于相同分割区的活动在同一工作流引擎上执行,这尤其适合于在一个企业内部,每个部门都部署一个工作流服务器的情况下。分割算法能够确保分割的正确性。

Statechart 具有定义严密的语义。例如,文[18]对 Statechart 定义了基于步(Step)的操作语义。因而,采用 Statechart 来建立工作流模型,从而使工作流模型能够进行验证。

### 3 基于规则的方法

ECA 规则,即事件-条件-动作规则,来源于主动数据库技术。ECA 规则已经被大量的工作流项目所采用<sup>[19,26]</sup>,它用来描述活动间的控制流关系。

在文[19]中,Bussler 用 ECA 规则来描述对执行实体的通知和同步策略。任务是执行实体的任务列表中的项,它们由执行实体选择一个或多个操作而执行完成。任务的执行必须根据由 ECA 规则规定的同步策略完成。ECA 规则的事件都是一些基本事件,这些事件可以与 WFMS 数据库中任务的状态有关。ECA 规则的条件表示与任务/执行实体关系、执行实体选择、任务完成状况等有关的完整性约束。其动作表示对数据库中任务状态和执行实体的任务列表的更改。

在 METEOR 项目中<sup>[20]</sup>,类似于 ECA 的规则用来表示任务间的状态和价值依赖关系。状态依赖规定了一个可控制任务变迁如何依赖于其它任务的可见状态。事件是隐含的,表示任务迁移到某个状态。规则中的条件可能是任务输出数据、工作流全局变量的表达式。规则中的动作则为一个任务迁移到新的状态。

G. Kappel 在 TriGSflow 工作流系统中<sup>[24,25]</sup>,采用 ECA 规则表示活动协调(即活动排序)、执行实体选择策略以及任务表管理策略。活动排序策略包括常见的 AND 连接、XOR 连

接等，它们都映射成 ECA 规则。执行实体选择策略用来选择当有多个合格的执行实体能够执行活动时，强制选择规则。例如，当有多个执行实体能够担当角色 A 时，可以选择具有最小负荷的执行实体以执行活动。另外，选择策略也可选择过去执行过此活动的执行实体。通常，一个执行实体所执行的任务具有时间约束，比如“活动 A 必须在 12:00 以前进行”。因而，任务管理策略用来进行任务的选择，以保证各个任务满足业务过程所需的时间约束。这些策略都用 ECA 规则表示。TriGSflow 采用了面向对象的工作流模型，例如，所有的活动都是 Activity 类的实例。因而，ECA 规则绑定于对象上。TriGSflow 系统将 ECA 规则与面向对象技术相结合。

A. Geppert 也采用了 ECA 规则来描述工作流中的控制流关系。在文[21]中，工作流过程被建模成提供服务的代理 (Broker) 的集合。代理对应于用户或软件执行实体，其用主动数据库 SAMOS 中的对象来实现。代理之间并不借助于任务表进行异步通信，而是通过请求或应答其它代理的服务而同步通信。服务对应于实现活动功能的应用。同样，活动间的控制结构，如 AND-join, OR-join 等，通过使用事件复合表达式而被映射成 ECA 规则。条件执行可能取决于前个活动的参数和它们对应的代理的环境值。事件是通过实现前个活动的服务完成时显示地触发。ECA 中的动作为请求对应于后续活动的代理的服务。活动间的时间约束，如活动的完成期限，也被当作控制流对待，并被映射成有时态事件触发的 ECA 规则。与 TriGSflow 系统不同，其工作流系统因而是一个由多个 Broker 组成的分布式执行系统。

A. Goh 将工作流技术与基于 STEP 的产品管理相结合<sup>[28]</sup>，运用于产品的需求、设计等业务流程中。同样，工作流规范映射到 ECA 规则上。此外，也提到了采用 ECA 规则可以灵活地支持过程改变、角色改变，从而使工作流系统具有更大的柔性。

在 WIDE 项目中，ECA 规则用来描述由时态事件、工作流内部事件和外部事件引发的工作流例外处理<sup>[31]</sup>。工作流内部事件指的是工作流变量更改、约束违例、任务取消或拒绝、执行实体的不可用等。条件可能是有关工作流模型变量或产品数据的表达式，也可以是时态表达式。动作可以是任务状态修改、例外处理例程。

从以上可以看出，ECA 规则不仅能够描述工作流模型中活动之间的协调关系，而且能够表达组织、角色之间的约束关系，如将任务分配到执行实体的策略等。此外，基于规则的方法能够支持过程的动态修改，角色的调整等，因而使工作流模型具有一定的灵活性，从而能够描述 Ad-hoc 类型的工作流模型。但 ECA 规则作为一种低层实现机制，缺乏图示化的手段和结构化构造工作流模型的方法。另外，当规则急剧增加时，对规则的处理将是一个更为繁琐的任务。ECA 规则也缺乏逐步求精的手段。就语义而言，ECA 规则本身必须就借助于其它方法来表示，如 Petri 网和时态逻辑。因而，很难对 ECA 规则表示的工作流模型进行形式化的验证。

#### 4 基于逻辑的方法

Davulcu 采用了并发事务逻辑 CTR (Concurrent transition Logic) 来表示工作流模型<sup>[32]</sup>。在 CTR 逻辑中，除了经典的逻辑连接符号  $\wedge, \vee, \rightarrow, \forall, \exists$  外，还增加了另外的逻辑连接符  $\oplus, |, \diamond, \odot$ 。它们分别表示顺序、并发、可能、单独执行。从而，工作流的控制流可以用这些逻辑符号组成的合式公式表示。应用 APPLY 编译算法能够将一个工作流图编译成 CTR 表示的公式。与基于图的方法相比，采用逻辑的一个优

点是，能够表达全局约束。这些全局约束也能够用逻辑来刻画，从而使工作流的描述和验证位于一个统一的框架中。但是基于逻辑的建模方法，由于其难度和缺乏方便的图示化手段，目前并没有成为工作流的主流建模方法之一。

#### 5 基于代数的方法

进程代数是一种刻画并发计算的一种工具，目前在工作流研究领域还没有得到广泛的研究和引起足够的重视。文[34]采用 LOTOS 来进行软件过程的建模。LOTOS 是一种描述通信的进程代数。由于软件过程和业务过程的相似性，因而也证明了进程代数可以用来刻画工作流模型。文[32]中也提到了这点。

同基于逻辑的方法一样，进程代数的方法具有严密定义的形式化语义。因而，这使工作流模型很容易进行验证。而进程代数优于逻辑的优点是，它提供了一个从低层子系统而复合系统的方法。其缺点是，缺乏图示化的手段。

#### 6 工作流建模方法小结

表1对前述的工作流建模方法进行了比较。在建模方法的可视化方面，基于图的方法都具有图示化的手段，如活动网络图、Petri 网、Statechart 图等，因而便于用户对过程具有直观的理解。在形式化语义方面，程序语言、活动网络图方法都缺乏严密定义的语义，而 Petri 网方法、逻辑方法和基于代数的方法都具有严密定义的语义，因而能对工作流模型进行形式化分析和验证。而基于 ECA 规则的方法其语义必须借助于 Petri 网或其它方法表达，因而难以对其进行建模。

表1 工作流建模方法

工作流建模方法	可视化	形式化语义	验证方法	描述范围	过程模块化	难易程度	典型系统
过程程序语言	×	×	×	过程模型	√	易	MOBILE APPL/A
基于图的方法	活动网络图	√	×	过程模型	不完全	一般	IBM FlowMark
	Petri 网	√	√可达图	过程、资源	不完全	难	INFCOM
	StateChart 方法	√	√Model Checking	过程功能组织	√	一般、图形紧凑	MENTOR
基于 ECA 规则	×	借助于其它方法	×	过程资源分配例外处理	×	较难	TriGSflow METEOR
基于逻辑的方法	×	√	√定理证明	过程	×	难	
基于代数的方法	×	√	√互模拟或逻辑	过程	√	难	

基于 Petri 网、ECA 规则和 statechart 方法，除描述过程模型外，还能表示其它方面的信息。基于 ECA 规则的方法具有较大的灵活性，因而还能够描述过程更改、例外处理、执行实体分配等方面的策略。基于 statechart 的方法可以表示系统的组织、功能、过程视图。而 Petri 网的方法也可以表示资源共享的情况。在过程的模块化方面，过程语言具有一些语言构

造以表示子过程,从而支持过程的重用。而活动网络图的方法,它仅支持块(Block),而并不支持子过程。Statechart 的方法通过层次分解从而很好支持子过程的重用。进程代数方法通常提供了一种从低层子系统复合成高层系统的方法,因而也支持子过程的构造。而 ECA 规则和逻辑的方法在这方面显示了不足,缺乏对子过程的支持和逐步求精的手段。在用户掌握的难易程度方面,通常 Petri 网的方法、逻辑方法和代数的方法对用户来说,具有较大的难度,而其它方法相对来说更为容易。

总之,目前的工作流产品中采用 Petri 网和 ECA 规则的方法比较多。Petri 网由于具有图示化的手段和严密定义的语义,能够对工作流模型进行形式化分析和验证,因而被广泛采用。但 Petri 网一个主要的缺点就是,图形过于复杂,节点过多,实际上普通用户较难掌握。而 ECA 规则的方法是一种较为理想的低层方式,但缺乏图示化的手段和逐步求精的手段。基于 Statechart 的方法不仅具有图示化的手段,而且具有严密定义的语义,能够对工作流模型进行分析和验证,并且能够建模信息系统的功能、组织、过程视图。

## 7 工作流模型的验证

工作流模型的正确执行是提高企业竞争力,改善企业服务质量的关键因素之一。包含逻辑错误的过程模型将导致丢失的货物、顾客的抱怨、利润的失去。过程模型中的一些缺陷也可能产生低的服务质量,并需要额外的修补措施。因而,在工作流模型执行以前,一个重要的要求之一就是能对工作流模型进行分析。总的说来,大概有以下三类分析方法:

1) 证实(Validation),即测试工作流运行是否和规定的模型一致。

2) 验证(Verification),即证明工作流模型是正确的。

3) 性能分析,即估算工作流的运行时间、服务质量、资源利用情况是否满足要求。

第一种和第三种方法,即证实(Validation)和性能分析,能够通过仿真进行。也就是说,通过一些大量的工作流实例,以检查工作流运行是否能很好地按设计要求完成。而第二种方法,即工作流模型的验证,必须在工作流执行以前,对模型

进行形式化的验证,以证明模型能100%的正确。因而,验证的方法更优于证实方法,因为经过验证正确的模型能够确保成功。

而验证问题在大多数商品化工作流产品中并没有涉及到,仅在少数几个研究项目中谈到<sup>[1,42]</sup>。例如,尽管 WIDE 项目在概念层采用了 WFDL 语言来描述工作流模型,其实现是基于 ECA 主动规则。这些规则能够由工作流规范自动或半自动地产生。但它并没有给出其形式化语义,因而,也不可能推理工作流规范的正确性。

Hofstede 首先谈到了工作流模型验证的复杂性问题。通过限制模型为无循环和无同步元素,他证明了终止性是一个多项式复杂度问题<sup>[42]</sup>。W. M. P Wander 通过把工作流模型表示成一种 Petri 网的变种—WF-net 网,从而利用经典 Petri 网的性质对工作流模型进行验证。他证明了一个工作流是有效的(Soundness)当且仅当其对应的 WF-net 网是活的,并且有界的,从而可以对 Petri 网的可达图等性质进行验证。此外,Wander 也确定了具有“良好结构”的静态结构约束<sup>[41]</sup>。值得一提的是,在 Mentor 项目中,工作流的正确性用时态逻辑来表示。而这些正确性主要是执行时的正确性,即动态约束。工作流模型用 Statecharts 来表示,而 Statecharts 与有限状态自动机密切相关。因而,Mentor 项目中采用了 SMC (Symbolic Model Checking)来验证工作流的正确性。SMC 是 Model Checking 的一种。Model Checking 是比定理证明更有效的一种模型验证方法,它能够验证给定的规范是否满足设计属性。本质上,Model Checking 验证一个给定的有限自动机是否是一个给定的时态逻辑公式的模型。目前,SMC 是最有效的 Model Checking 方法,它是基于用 OBDD 紧凑、符号表示有限状态自动机<sup>[12]</sup>。

总之,较之工作流仿真而言,工作流验证更未引起足够的重视。只有少数几个项目对其进行了研究。在表2中,对目前的工作流验证方法进行了总结。Petri 网的方法仍然存在一些不足,它只能描述通用的结构属性,如不存在死锁等。并没有考虑每个特定任务的语义,这一点在文[40]中也提及。相比之下,Mentor 项目中采用 statechart 的建模方法,而用时态逻辑来表示属性,它能够表示与特定应用有关的语义。

表2 常见的工作流验证方法对比

研究人员/研究项目	工作流建模方法	验证方法	能够验证的属性(正确性)	工具支持
Sadiq	基于图的方法	基于图论算法	通用属性	专门
Wander Alast	基于 Petri 网的方法	Petri 网的活性、有界性	通用属性	专门/通用
Mentor	基于 Statecharts	Symbolic Model Cheking	通用属性,与任务语义相关的特定属性	通用如 SMV, SPIN 等

**总结** 一个比较理想的工作流建模方法应该满足以下条件:

1) 具有图示化的界面,建模用户容易掌握。

2) 建模方法或语言本身是可执行的,即可以作为工作流引擎的实现机制。例如,一般基于规则或基于状态的方法都可以作为低层实现机制。

3) 具有形式化的语义。也就是说,其语义可以用 Petri 网、逻辑、进程代数等表示,从而可以对模型进行验证、推理。

从以上可以看出,在前述的建模方法中,基于 Petri 网的建模方法和基于 statechart 的方法基本上能满足以上要求。相比之下,基于 statechart 的方法由于图形简明、紧凑,而以 statechart 为基础的 UML statechart 已成为面向对象的标准,因而采用 statechart 的建模方法将会在工作流建模中占有重要的位置。

另外一种比较通用的办法就是采用一种“混合”方法(Hybrid),即在高层面向用户方面采用具有图示化的建模手段或工作流建模语言,而在低层(或内部)采用一种或多种便于实现或具有严格语义、便于验证的方法,从而可以将高层描述方法转化成低层方法,便于模型验证和程序实现。从表1中可以看出,一般基于图的方法具有良好的图示化界面,而基于逻辑、进程代数的方法具有严密定义的语义,基于规则和方法便于作为内部实现机制。因此,一个完整的工作流系统往往是上述几种方法的综合。例如,WIDE 项目在概念层采用了 WFDL 语言(WorkFlow Description Language)来描述工作流规范,而在低层实现采用基于 ECA 主动规则的方法。因而,我们可以预料:在实际的工作流产品中,越来越多的产品采用这种“混合”方法,从而不仅使普通用户容易使用、建模,而且方便实现、形式化分析和验证。

## 参考文献

- 1 Georgakopoulos D, Hornick M F, Sheth A P. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 1995, 3(2): 119~153
- 2 Workflow Management Coalition, WPMC-TC-1003, 19-Jan-95, 1. 1, The Workflow Reference Model, 1995
- 3 Stephen J H, Yang. a Petri-net-based approach for workflow and process automation. *International Journal on Artificial Intelligent Tools*, 1999
- 4 范玉顺. 基于工作流的企业过程的建模和仿真技术研究. *清华大学学报*, 2000, 40(1)
- 5 Oberweis A, et al. INCOME/WF - a Petri net based approach to workflow management. In Krallmann, H. ed. *Wirtschaftsinformatik '97*, Berlin, Germany, Physica-Verlag, Heidelberg, Germany, 1997. 557~580
- 6 Ellis C, Nutt G. Modeling and Enactment of Workflow Systems, in: *Application and Theory of Petri Nets*, Lecture Notes in Computer Science 691. Springer-Verlag, 1993. 1~16
- 7 van der Aalst W M P. Modelling and analysing workflow using a Petri-net based approach. In: *Proc. 2nd Workshop on Computer-Supported Cooperative Work, Petri nets and related formalisms*, 1994. 31~50
- 8 van der Aalst W M P. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 1998, 8(1): 21~66
- 9 van der Aalst W M P. Three good reasons for using a Petri-net-based workflow management system. In *Information and process integration in enterprises: Rethinking documents*, Kluwer Academic, Norwell, MA, 1998. 161~182
- 10 Adam N R, et al. Modeling and Analysis of Workflows Using Petri Nets. *Journal of Intelligent Information Systems (JIIS)*, 1998, 10: 131~158
- 11 Choi I, Park C, Lee C. Task-Net: Transactional Workflow Model based on Colored Petri Net. *European Journal of Operational Research*, 2002, 136: 383~402
- 12 Muth P, et al. Enterprise-wide Workflow Management Based on State and Activity Charts, NATO Advanced Study Institute on Workflow Management Systems and Interoperability, 1997
- 13 Alonso G, Mohan C. Workflow Management: The Next Generation of Distributed Processing Tools, Chapter 2 in *Advanced Transaction Models and Architectures*. In: S. Jajodia, L. Kerschberg, eds. Kluwer Academic Publishers, 1997
- 14 Curtis B, Kellner M I, Over J. Process Modeling. *Communications of ACM*, 1992, 35(9): 75~90
- 15 Harel D. Statecharts: A Visual Formalism for Complex System. *Science of Computer Programming*, 1987, 8(3): 231~274
- 16 Harel D, Naamad A. The State Semantics of Statecharts. *ACM Transactions on Software Engineering and Methodology*, 1996, 5: 293~333
- 17 Harel D, et al. STATEMATE: A Working Environment for the Development of Complex Reactive Systems. *IEEE Transactions on Software Engineering*, 1990, 16(4)
- 18 Harel D, Pnueli A, Schmidt J P, Sherman R. On the Formal Semantic of Statecharts. In: *Proc. of IEEE Symposium on Logic in Computer Science (LICS 1987)*, 1987. 54~64
- 19 Bussler C, Jablonski S. Implementing Agent Coordination for Workflow Management Systems Using Active Database Systems. In: *Proc. Research Issues in Data Engineering*, (Houston, TX, February 1994). IEEE Computer Society Press, 1994
- 20 Krishnakumar N, Sheth A. Managing Heterogeneous Multi-System Tasks to Support Enterprise-Wide Operations. *Distributed and Parallel Databases*, 1995, 3(2): 155~186
- 21 Geppert A, Kradolfer M, Tombros D. Realization of Cooperative Agents using an Active Object-Oriented Database System. *Rules in Database Systems*, 1995. 327~341
- 22 Tombros D, Geppert A, Dittrich K R. The Broker/Services Model for the Design of Cooperative Process-Oriented Environments: [Technical Report 97. 06]. Department of Computer Science, University of Zurich, Switzerland, 1997
- 23 Geppert A, Kradolfer M, Tombros D. Workflow Specification and Event-Driven Workflow Execution
- 24 Kappel G, Rausch-Schott S, Retschitzegger W. Coordination in Workflow Management Systems - A Rule-based Approach. Springer LNCS 1364, 1998
- 25 Kappel G, et al. TriGS - Making a Passive Object-Oriented Database System Active. *Journal of Object-Oriented Programming (JOOP)*, 1994, 7(4): 40~51
- 26 Kappel G, et al. Workflow Management based on Objects, Rules and Roles. *IEEE Data Engineering Bulletin, Special Issue on Workflow Systems*, 1995, 18(1): 11~18
- 27 Kappel G, Raush-Shott S, Retschitzegger W. From rules to rule patterns. In *caise96*, Crete, Greece. Springer-Verlag, Berlin, 1996. 99~115
- 28 Goh, koh Y K, Domazet D S. ECA rule-based support for workflows. *Artificial Intelligence in Engineering*, 2001, 15: 37~46
- 29 Casati F, Ceri S, Pernici B, Pozzi G. Conceptual Modeling of Workflows. In: *Proc. Object-Oriented and Entity Relationship Approach '95 Intl. Conf.* Springer Verlag, 1995
- 30 Casati F, Ceri S, Pernici B, Pozzi G. Deriving Active Rules for Workflow Enactment. In: *Proc. 7th Intl. Conf. on Database and Expert System Applications*, Springer Verlag, 1996
- 31 Casati F, Grefen P, Sanchez G. WIDE - A Distributed Architecture for workflow Management. In: *Proc. 7th Intl. Workshop on Research Issues in Data Engineering (Birmingham, England, April 1997)*
- 32 Davulcu H, Kifer M, Ramakrishnan C R, Ramakrishnan I V. Logic Based Modeling and Analysis of Workflows. In *ACM Symposium on Principles of Database Systems*; Seattle, Washington, ACM Press, 1998
- 33 Liu L, Meersman R. The building blocks for specifying communication behavior of complex objects: an activity-driven approach. *ACM Transactions on Database Systems*, 1996, 21(2): 157~207
- 34 Saeki M, Kaneko T, Sakamoto M. A Method for Software Process Modeling and Description Using LOTOS. In: *Proc. of the First Intl. Conf. on the Software Process*, IEEE Computer Society, Oct. 1991. 90~104
- 35 Sutton S, Heimbigner D, Osterweil L J. APPL/A; A Language for Software Process Programming. *ACM Transactions on Software Engineering and Methodology*, 1995, 4(3)
- 36 Jablonski S, Bussler C. Workflow Management Modeling Concepts, Architecture and Implementation. International Thomson Computer Press, 1996
- 37 Fernstrom C. Process Weaver: Adding Process Support to UNIX. In: *Proc. of the 2nd Intl. Conf. on the Software Process - Continuous Software Process Improvement*, IEEE Computer Society Press, 1993
- 38 van der Aalst W M P. Verification of Workflow Nets. *Application and Theory of Petri Nets 1997*, Vol 1248 of Lecture Notes in Computer Science, pages 407~426. Springer-Verlag, Berlin, 1997. Editors P. Azema and G. Balbo
- 39 Wodtke D, Weikum G. A formal foundation for distributed workflow execution based on state charts. In: F. N. Afrati, P. Kolaitis, ed. *6th Intl. Conf. on Database Theory (ICDT)*, volume 1186 of Lecture Notes in Computer Science, Springer, 1997. 230~146
- 40 Fu Xiang, Bultan T, Hull R, Su J. Verification of Vortex Workflows. *Proc. TACAS 2001*, LNCS vol. 2031
- 41 van der Aalst W M P. Verification of Workflow Nets. *Application and Theory of Petri Nets 1997*, Vol 1248 of Lecture Notes in Computer Science, pages 407~426. Springer-Verlag, Berlin, 1997. Editors P. Azema and G. Balbo
- 42 ter Hofstede A H M, Orlowska M E, Rajapakse J. Verification Problems in Conceptual Workflow Specification: [Technical Report No. 363]. University of Queensland, April 1996