

# 基于 UML-Statecharts 的工作流控制结构的验证\*

陆公正<sup>1</sup> 戎 玫<sup>2</sup> 张广泉<sup>1,3</sup>

(苏州大学计算机科学与技术学院 苏州 215006)<sup>1</sup> (暨南大学深圳旅游学院 深圳 518053)<sup>2</sup>

(重庆师范大学数学与计算机科学学院 重庆 400047)<sup>3</sup>

**摘 要** 工作流模型验证成为工作流的重要研究领域之一。控制结构的正确性是工作流过程所需达到的最基本要求,本文着重控制结构方面的验证。本文使用 UML-Statecharts 建立控制结构模型,使用时序逻辑表示工作流控制结构需要满足的性质。给出了一个定理并进行了证明,基于定理给出了一个验证完全性的算法,对于工作流语义相关性质的验证给出了一模型检测算法。

**关键词** 工作流, UML-Statecharts, 时序逻辑, 完全性, 模型检测

## UML-statecharts Based Verification of Control Flow of the Workflow

LU Gong-Zheng<sup>1</sup> RONG Mei<sup>2</sup> ZHANG Guang-Quan<sup>1,3</sup>

(School of Computer Science and Technology, Soochow University, Suzhou 215006)<sup>1</sup>

(Shenzhen Tourism College, Jinan University, Shenzhen 518053)<sup>2</sup>

(School of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047)<sup>3</sup>

**Abstract** Verification of workflow model has been one of the important research fields of workflow. The correctness of the control flow is the basic requirement that the workflow process must satisfy. In this paper, we emphasize on the verification of the control flow. We construct the control flow model by UML-Statecharts, and express that the property the control flow must satisfy by temporal logic. Then we give a theorem and prove it, also according to it give an algorithm of verification of soundness. For the verification of the property about workflow semantics we also give an algorithm.

**Keywords** Workflow, UML-Statecharts, Temporal logic, Soundness, Model checking

## 1 简介

工作流管理联盟(WFMC)给出了工作流的标准定义<sup>[1]</sup>;工作流是一类能够完全或部分自动执行的经营过程,根据一系列过程规则、文档信息和任务能在不同的执行者之间传递、执行。对工作流进行管理、协调和监控的工作流管理系统目前得到广泛的应用,并且与它相关的理论也在迅速发展。

工作流模型验证已经成为工作流的重要研究领域之一。由于控制结构可以从工作流的其它方面抽象出来<sup>[2,3]</sup>,我们能够单独对其验证。控制结构的验证包括两个方面:一方面是工作流的控制结构是否是完全的(sound),例如工作流模型中是否存在死锁;另一方面是与工作流语义相关的,例如工作流模型是否满足某一所期望的性质。控制结构的正确性只是工作流过程所需达到的最基本的要求。

目前出现了很多工作流模型的验证方法,应用最广泛的是基于 Petri 网的方法。但是由于文[4]中提到的原因,本文使用基于 UML-Statecharts 的方法。它同样具有严格的形式化语义,而且它可以认为是有限状态图、正交性、深度和广播通信机制的组合,很适合工作流过程的建模,最后它可以应用模型检测理论进行验证。

模型检测(model checking)是一种重要的自动验证技术,主要通过显式状态搜索或隐式不动点计算来验证有穷状态并

发或实时系统的模态/命题性质。近来出现了一些对 Statecharts 进行模型检测的研究。文[5]中把扩展层次自动机(EHA)作为中间模型,把 Statecharts 转换为 SMV 或 SPIN 的输入语言。文[6]把 UML-Statecharts 转换为 SPIN 的输入语言。

首先简单介绍一下 UML-Statecharts 以及它的操作语义,然后给出控制结构所需满足的性质以及形式化表示;接着给出了一个定义并进行了证明,基于定理给出了一个验证完全性的算法;对于工作流时序性质的验证给出了一个模型检测算法;最后是结论。

## 2 UML-Statecharts

### 2.1 UML 状态图

UML 状态图是一种扩展的有限状态机,在传统的有限状态机上增加了层次、并发和通信机制,是一种强大、灵活的状态迁移图。

UML 状态图所刻画的对象状态分为基本状态、复合状态和伪状态。非并发的复合状态称为 OR 状态,并发的复合状态称为 AND-状态。复合状态可以包括多个子状态,用于表示层次结构和并发。伪状态包括初始、终止、同步、join、fork 和 choice 等。UML 状态图的迁移包括源状态、目标状态、激发事件、布尔条件和动作。UML 状态图中用方框表示

\* 本课题得到中国科学院计算机科学国家重点实验室开放课题(编号 SYSKF0303)、重庆市科学技术研究项目(编号 040803)资助。陆公正 硕士研究生,主要研究方向为形式化方法;张广泉 教授,主要研究方向为形式化方法、软件体系结构。

状态,用有向箭头表示状态间的迁移,方框内包含的方框是该状态的子状态,实心圆表示初始状态,空心圆表示终止状态。

workflow中的顺序结构可以用 OR-状态间的顺序执行表示,并发结构可以用 AND-状态中的多个子状态表示,选择结构可以通过一个源状态有多个目标状态表示,循环结构可以通过状态的迁移指向它的前继状态表示。下面给出一个顾客投诉的例子(如图1)。

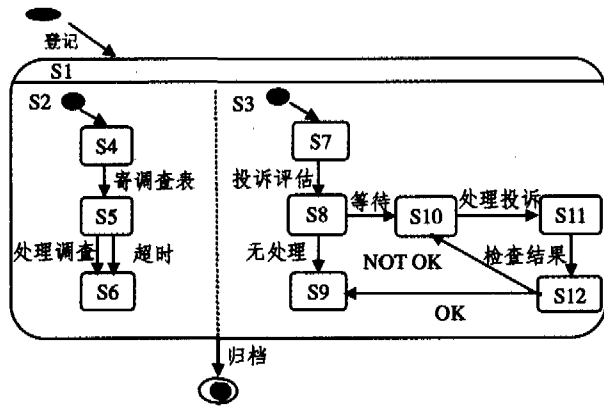


图1 顾客投诉 workflow 模型

### 2.2 UML 状态图的操作语义<sup>[7]</sup>

对 UML 状态图的模型检测通常需要先把它转换为相应的 EHA。EHA 由简单的顺序自动机组成,状态通过精化函数映射到一组对其进行细化的(并发)顺序自动机。

**定义 2.1(顺序自动机)** 顺序自动机 A 是一个四元组  $(\sigma_A, S_A^i, \lambda_A, \delta_A)$ ,  $\sigma_A$  是用穷状态集合,  $S_A^i$  是初始状态,  $\lambda_A$  是有穷的迁移标记集合,  $\delta_A \subseteq \sigma_A \times \lambda_A \times \sigma_A$  是迁移关系。

**定义 2.2(扩展层次自动机)** EHA H 是一个五元组  $(F, E, \rho, A_0, V)$ , F 是有穷的顺序自动机集合,  $\forall A_1, A_2 \in F, \sigma_{A_1} \cap \sigma_{A_2} = \emptyset$ ; E 是有穷事件集合; V 是变量集合;  $\rho: \bigcup_{A \in F} \sigma_A \rightarrow 2^F$  是精化函数,给出 F 的一个树型结构满足:(1)存在唯一的根自动机  $A_0 \in F$ ,使得不存在  $s \in \bigcup_{A \in F} \sigma_A, A_0 \in \rho(s)$ ;(2)每个非根自动机恰有一个父状态,对于  $\forall A \in F \setminus \{A_0\}, \exists s \in \bigcup_{A' \in F(A) \setminus \{A\}} \sigma_{A'}, A \in \rho(s)$ ;(3)不存在环路:  $\forall S \subseteq \bigcup_{A \in F} \sigma_A, \exists s \in S, s \cap (\bigcup_{A \in \rho(s)} \sigma_A) = \emptyset$ 。

UML-Statecharts 转换为 EHA 时发生非层间迁移。为了不改变语义,需要对迁移标记进行扩充,添加受限源状态 sr 和确定目标状态 td,迁移标记变为五元组  $(sr, ev, g, ac, td)$ ,其中 sr 将迁移使能限制在源状态的一个格局中, ev 是触发事件, g 是布尔条件, ac 是动作, td 决定了迁移使系统进入目标状态时有哪些子状态也同时进入。

层次自动机的全局状态由格局表示,它由包含的顺序自动机的某些局部状态组成。ConfH 表示由 EHA H 中的所有有效格局组成的集合。

**定义 2.3(格局)** H 的一个格局是一个集合  $Conf \subseteq \bigcup_{A \in F} \sigma_A$  使得:(1)  $\exists s \in \sigma_{A_0}$  满足  $s \in Conf$ ;(2)  $\forall s, A$ , 若  $s \in Conf$  且  $A \in \rho(s)$ , 则  $\exists s' \in A, s' \in Conf$ 。

在 UML-Statecharts 中,操作语义中的状态称为状况(status),而迁移就代表 RTC 步。每个状况由格局和当前的环境组成,环境包括当前事件队列中存在的事件及其顺序和当前所有变量取值,分别称为事件环境和变量环境。对于事件集合 X,用  $\Theta X$  表示在 X 上某种类型事件队列的所有可能结构。对于变量集合 Y,  $\Omega Y$  表示对 Y 中变量的所有可能赋

值。

下面用 LTS 对经过扩充的 EHA 定义操作语义。

**定义 2.4(EHA 的操作语义)** 一个 EHA H 的操作语义是一个 LTS  $T_S = (S, S_0, L, \rightarrow)$ , 其中  $S = Conf_H \times \Theta X \times \Omega Y$  是状况集合,  $S_0 = (C_0, E_0, V_0)$  是初始状况,  $L: S \rightarrow 2^{AP}$  给每个状况标注一组原子命题,  $\rightarrow S \times S$  是迁移关系。

$T_S$  中的一个迁移是状态图中通过选择最大无冲突迁移集所完成的一个 RTC 步<sup>[7]</sup>, 关系  $\rightarrow$  通过一个演绎系统进行定义,由三条语义规则给出:前进规则、组合规则和暂停规则,具体见文<sup>[7]</sup>。

### 3 控制结构所需满足的性质

#### 3.1 完全性

在工作流验证时,使用完全性作为控制流的正确性标准。 workflow 过程是完全的,如果它满足如下条件<sup>[2]</sup>:

(1)根据过程处理的案例总是可能完成的,这一条件保证了没有死锁和活锁。

(2) workflow 过程发出案例完成的信号不可能仍有任务在执行。

(3)对于每一个任务,在 workflow 执行过程中它都有执行的机会,这一约束意味着在 workflow 过程中每一个任务都有有意义的的作用。

workflow 过程是完全的,当且仅当:

(1)对于从初始状态可达的状态 s,都存在从 s 到终止状态的迁移序列。形式地:

$$\forall s \in \bigcup_{A \in F} \sigma_A (i \xrightarrow{*} s) \Rightarrow (s \xrightarrow{*} o)$$

(2) workflow 过程中没有死任务。形式地:

$$\forall i \in \bigcup_{A \in F} \sigma_A, \exists s, s' (i \xrightarrow{*} s \xrightarrow{t} s')$$

#### 3.2 语义相关的性质

控制结构验证除了验证它是否是完全的之外,还需验证与工作流语义相关的性质,即它是否满足某些所期望的性质,这些性质用时序逻辑表示。

时序逻辑是对命题逻辑的扩展,它引入了与时间相关的算子,例如  $\square$  (必然算子)、 $\diamond$  (终于算子)、 $\bigcirc$  (下一时刻算子) 和  $u$  (直到算子)。本文所使用的是 CTL(计算树逻辑),它的语法和语义<sup>[8]</sup>如下。

**定义 3.1(CTL 语法)** CTL 的结构是三元组  $M = (S, L, \lambda)$

- S 是格局集

- L 是  $S \rightarrow 2^{AP}$  是一标记函数,它赋给每一个格局原子命题集合为真

- $\lambda$  是迁移集

与工作流语义相关的性质都可以使用下面的 CTL 公式表示,其它的公式都可以转化到这几种形式。

**定义 3.2(CTL 公式)**  $\phi ::= p \mid \text{false} \mid \phi_1 \rightarrow \phi_2 \mid \exists \bigcirc \phi_1 \mid \exists (\phi_1 u \phi_2) \mid \forall (\phi_1 u \phi_2)$ , 其中  $p \in AP, \phi_1, \phi_2$  是 CTL 公式。

**定义 3.3(CTL 语义)** 对于 CTL 结构 M、格局 C 和公式  $\phi, (M, C) \models \phi$  定义如下:

$$C \models p \text{ 当且仅当 } p \in L(C)$$

$$C \not\models \text{false}$$

$$C \models \phi_1 \rightarrow \phi_2 \text{ 当且仅当 } C \not\models \phi_1 \text{ 或者 } C \models \phi_2$$

$$C \models \exists \bigcirc \phi \text{ 当且仅当对于某一 } (C, C') \in \lambda \text{ 的格局 } C' \in S \text{ 有 } C' \models \phi$$

$C \vdash \exists (\phi_1 \wedge \phi_2)$  当且仅当对于某一有  $C=C_0$  的路径  $(C_0, C_1, \dots)$ , 对于某一  $i \geq 0, C_i \vdash \phi_2$  且对于  $0 \leq j < i$  有  $C_j \vdash \phi_1$

$C \vdash \forall (\phi_1 \wedge \phi_2)$  当且仅当对于每一有  $C=C_0$  的路径  $(C_0, C_1, \dots)$ , 对于某一  $i \geq 0, C_i \vdash \phi_2$  且对于  $0 \leq j < i$  有  $C_j \vdash \phi_1$

#### 4 控制结构验证

##### 4.1 全局可达状态迁移图

在进行控制结构验证之前,我们先要生成工作流过程的全局可达状态迁移图。这里,我们不考虑变量环境。

全局可达状态迁移图  $G=(Conf_H, R, L, Conf_0)$  是一个 kripke 结构。其中  $Conf_H$  为格局集,  $(conf_1, conf_2) \in R$  表示存在从  $conf_1$  到  $conf_2$  的迁移,  $L: Conf_H \rightarrow 2^{AP}$  为每个格局标注一组原子命题,  $conf_0$  是初始格局。假设  $R$  中的迁移互不相同。

生成全局可达迁移图的算法如下:

- (1)  $Conf_0 = \{i\}$ , 把  $i$  的所有活跃子状态的初始状态加入  $Conf_0$  中。
- (2) 从事件队列中删除当前触发事件, 得到可以被当前事件使能的所有迁移集合。
- (3) 检查迁移集合中是否存在冲突的迁移。如果存在, 则根据迁移间的优先关系进行触发。
- (4) 根据迁移关系到达所有可达状态。如果当前触发状态没有其它迁移, 则从当前格局中删除, 否则保留。把所有可达状态中与当前格局中不同的状态加入当前格局, 生成新的格局; 如果生成的新格局中没有新状态, 则算法终止, 生成错误信息。
- (5) 如果格局中的状态都为终止状态, 则算法结束, 否则跳到(2)。

图 2 为图 1 的全局可达迁移状态图(省去迁移标记)。

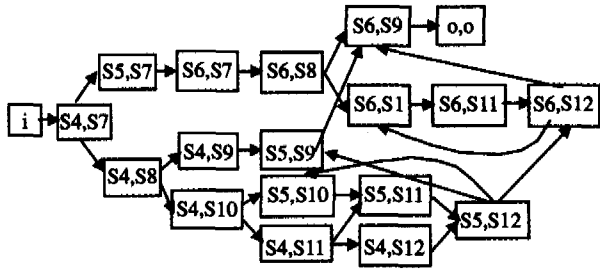


图 2 图 1 的全局可达状态迁移图

##### 4.2 完全性的验证

验证 UML-Statecharts 建立的工作流模型的是完全的, 我们可以通过验证根据它生成的全局可达状态迁移图是完全的。

**定理 4.1** 工作流模型是完全的, 当且仅当全局可达状态迁移图是完全的。

**证明:** 先证必要性。工作流模型是完全的, 那么它的所有状态都可以到达终止状态。因为  $Conf_H \subseteq \bigcup_{A \in PA} A$ , 那么全局可达状态迁移图中的所有状态都可以到达终止状态。工作流模型是完全的, 说明不存在死任务, 而全局可达状态迁移图是根据工作流模型生成的, 生成过程中只触发使能的迁移, 所以它也不存在死任务, 全局可达状态迁移图是完全的。

再证充分性。假设工作流模型中存在不可达的状态, 即工作流过程的完全性条件(1)中的条件部分为假, 那么条件(1)为真。全局可达状态迁移图中的状态是所有可达状态, 它是完全的, 那么其中的所有状态都可以到达终止状态, 所以条件(1)成立。假设工作流模型中存在死任务  $t$ , 它是状态  $s$  和  $s'$  之间的迁移关系,  $i$  可以到达状态  $s$ 。如果  $s$  只存在唯一迁移  $t$ , 那么在全局可达状态迁移图中存在从初始格局到包含状态  $s$  的格局的路径。但由于  $s$  只有唯一一个迁移  $t$ , 因此它

不存在达到终止状态的路径, 与全局可达状态迁移图是完全的矛盾; 如果  $s$  还存在使能迁移  $t'$ , 根据公平性,  $t$  最终使能。如果全局可达状态迁移图是完全的, 那么工作流模型中不存在死任务, 所以工作流模型是完全的。

下面给出验证可达迁移状态图完全性的算法。

```

Procedure BFScheck(v)
//已访问的结点都标上 visited(i)=1, 初始时 visited 都设为 0//
visited(v)=1; u=v
P, Q 初始化为空; // P 为使能迁移集合, Q 为未检测结点的队列//
loop
for 邻接于 u 的所有结点 w do
if u 到 w 的迁移 t 使能 then P=P∪{t} endif
if visited(w)=0 then 把 w 加入到 Q 的尾部 visited(w)=1 endif
repeat
if Q 为空 then return endif
从 Q 中取一个未检测的结点
repeat
ENDBFS.
Function checksound()
BFScheck(Conf_0)
//如果每个格局都访问到, 并可以到达终止格局, 而且每个迁移都使能, 那么它是完全的//
if all visited(Conf)=1 and P=R then return YES else return NO
endif;
ENDchecksound.
    
```

根据算法可得图 2 是完全的, 再根据定理 1, 那么图 1 的工作流模型是完全的。

##### 4.3 语义相关性质的验证

这里我们使用 CTL 表示工作流的时序逻辑性质。工作流的 UML-Statecharts 模型满足 CTL 公式  $F$  当且仅当它所对应的全局状态迁移图  $G$  满足  $F$ , 而  $G \vdash F$  iff  $Conf_0 \vdash F$ 。

下面给出模型检测算法。算法的基本思想是为每个格局  $Conf$  标注在它为 true 的  $F$  的子公式, 如果最后以公式  $F$  标注初始格局  $Conf_0$ , 那么工作流模型满足性质  $F$ 。其中  $label(Conf)$  表示为格局  $Conf$  标注命题集合,  $sub(F)$  表示  $F$  的所有子公式,  $G, Conf \vdash F$  iff  $F \in label(Conf_0)$ 。

```

Function checkCTL()
for each  $\phi \in sub(F)$  do
Case  $\phi$ 
 $\phi \in AP$ ; skip
 $\phi = f$ ; for each  $Conf \in Conf_H$  do
if  $f \notin label(Conf)$  then  $label(Conf) = label(Conf) \cup \phi$  endif
repeat
 $\phi = f \vee g$ ; for each  $Conf \in Conf_H$  do
if  $f \in label(Conf)$  or  $g \in label(Conf)$  then  $label(Conf) = label(Conf) \cup \phi$ 
endif
repeat
 $\phi = \exists \phi$ ; for each  $Conf_1, Conf_2 \in Conf_H$  do
if  $(Conf_1, Conf_2) \in R$  and  $label(Conf_2) = f$  then
 $label(Conf_1) = label(Conf_1) \cup \phi$ 
endif
repeat
 $\phi = \exists (f u g)$ ; for each  $Conf_i, Conf_j \in Conf_H$  do
if there exists a path from  $Conf_i$  to  $Conf_j$  and  $label(Conf_j) = g$  and  $label(Conf_i) = f$  then  $label(Conf_i) = label(Conf_i) \cup \phi$  endif
repeat
 $\phi = \forall (f u g)$ ; for each  $Conf_i, Conf_j \in Conf_H$  do
for every path from  $Conf_i$  to  $Conf_j$  do
if  $label(Conf_j) = g$  and  $label(Conf_i) \neq f$  then
 $label(Conf_i) = label(Conf_i) \cup \phi$ 
endif
repeat
if  $F \in label(Conf_0)$  then return YES else return NO endif
ENDcheckCTL.
    
```

(下转第 177 页)

格式中,旧版本中的部分格式调整到了新基本格式中,规范格式的覆盖范围更小了,但是语种特性却更加突出了。

### 4.3 违例格式

违例格式的设计体现了语言交流过程中交流双方在相邻 GBK 边界清楚的情况下省略标记的语言特性。虽然这种语言特性体现了自然语言天然抽象表达的特性,对人的交流不会存在困难,但是对于计算机理解自然语言来说则是一个大的困扰,好在标记的省略只特定于部分的特征语义块或者广义对象语义块中,只要根据新定义的违例格式知识结合相关的句类知识后就有效地解决这一困扰。新版本违例格式的编码在基本格式定义变化后进行缩减,采用了编码更易于记忆的设计原则,本质定义上同旧版本违例格式未有太大的变化。

从例句中可以看出,这种结构更多见于古汉语的习惯表达和句脱一级的局部表达习惯中,英汉都存在,但语句一级的表达不常见。

### 4.4 省略格式

省略格式不同于违例格式中边界标记的省略,它是对数学表示式中主语义块的整体省略,这种情况多见于句群中小句对前面小句中主语义块省略的情况。新旧版本省略格式的定义没有发生改变,具体的格式编码随着前 3 种格式的变化而相应变化。

例句中没有发生变化,只是在具体编码过程中,由于例 13 原是在规范格式基础上省略变为了在基本格式基础上变化,代码发生了相应的变化(! 111->! 09)。

## 5 新格式存在的问题

由于旧格式对语义块“时态、形态和格”变化采取置之不理的态度,可以很容易列出 2 块句、3 块句甚至 4 块句的排列中统计出格式的完备集合。新格式充分考虑语义块“时态、形态和格”变化,造成数学排列列出格式完备集合总数巨大,并且这也违背了自然语言格式分布不均的使用习惯特点,所以

新格式一般不列出代码表的完备集合,仅列出了部分常见的编码,具体代码由使用者在规则基础上自行完成。

新格式代码在旧基本格式基础上融入了大量旧规范和违例格式,虽然从语言表达习惯上强烈体现了汉英的语种特性,但是显然减弱了旧基本格式与句类表示式的天然联系性,导致基本格式! 0 作为英汉两种语言最习惯采用的格式的突出性明显减弱,这不能不说是由“鱼与熊掌不可兼得”所带来的矛盾。

**结语** HNC 理论是一个描述语言概念空间的符号体系,这个符号体系的一部分就是句类基元符号体系,可大体对应着自然语言中的语句,HNC 理论通过格式代码和句类代码相互配合能对语言空间中的各种复杂语言现象进行解释。语句格式用来表述语句中主语义块在句中的出现顺序。由于研究的深入,进一步加深了对句类和格式的理解,格式定义和编码也相应地发生了新的变化。变化后的新格式突出体现了以汉英为代表的语言的习惯表达、语法范畴“格、时、体、态、式”的综合运用。本文就语句格式的变化进行了系统的说明,详细阐述了新旧格式之间定义和编码的差异,从而有利于新格式的全面推广。不当之处还请读者不吝指出。

### 附录 相关的标记符号:

|| 语义块边界标记符;| 句脱中语义块的间隔符号;  
{ } 原型句脱标记符;<> 要素句脱标记符;\ / 包装句脱标记符;^ 语义块指示符;& 并联连接符。

## 参考文献

- 1 胡树鲜. 现代汉语语法理论初探. 中国人民大学出版社,1990
- 2 黄曾阳. HNC(概念层次网络)理论. 北京:清华大学出版社,1998
- 3 黄曾阳. 语言概念空间的基本定理和数学物理表达式. 北京:海洋出版社,2004
- 4 张全,萧国政. HNC 与语言学研究. 武汉:武汉理工大学出版社,2001

(上接第 161 页)

验证性质  $V \diamond (S6, S9)$ , 先转换为公式  $V (\text{true } u (S6, S9))$ , 根据模型检测算法,性质成立。

**结论** 把 UML-Statecharts 用于工作流的建模和验证已经成为当今的趋势,目前已经提出了它的基于 Statemate 的操作语义和基于步(Step)的操作语义,成为工作流验证的基础。本文得出了用 UML-Statecharts 建立的工作流模型是完全的当且仅当根据步语义生成的全局可达状态迁移图是完全的结论,这样简化了工作流完全性的验证。并且在工作流语义相关性验证中,使用了模型检测理论,它通过模型检测算法自动地验证时序性质,使验证过程有效而快速。但是 UML-Statecharts 的分析技术还不够成熟和完善,需要我们不断改进和发展,使其更有效地用于工作流的建模和验证。

## 参考文献

- 1 Hollingsworth D. Workflow Management Coalition-The Workflow Reference Model [M]. Workflow Management Coalition,

- 1995
- 2 Verbeek H M W, et al. Diagnosing Workflow Process Using Woflan [J]. Computer Journal,2001,44(4):246~279
- 3 van der Aalst W M P. Workflow Verification; Finding Control-Flow Errors Using Petri-Net-Based Techniques. In: van der Aalst W, et al, eds. Business Process Management, LNCS1806, Springer-Verlag,2000. 161~183
- 4 Oren E, Haller A. Formal Frameworks for Workflow Modelling. In:DERI-Digital Enterprise Research Institute [R],2005. 1~20
- 5 Bhaduri P, Remash S. Model Checking of Statechart Models-Survey and Research Directions [J]. CoRR cs. SE/0407038,2004
- 6 Latella D, et al. Automatic Verification of a Behavioural Subset of UML Statechart Digrams Using the SPIN Model-checker. Formal Aspects of Computing, 1999, 11: 637~664
- 7 Dong W, et al. Model Checking of UML State Machines [J]. Computer Engineering and Science,2001,23(6):7~11
- 8 Alur R,Courcoubetis C, Dill D L. Model-checking in dense real-time [J]. Information and Computation,1995,104:2~34