

# 基于ASM描述共享网络的实时任务可执行概率设计<sup>①</sup>

张冬松, 金士尧, 陈芳园, 李超

(国防科学技术大学 并行与分布处理国防科技重点实验室, 长沙 410073)

(国防科学技术大学 计算机学院, 长沙 410073)

**摘要:** 利用抽象状态机(ASM), 形式化描述了共享网络中实时任务接纳控制机制, 提出了实时任务可执行概率设计过程的ASM模型, 利用Spec Explorer给出了分析该模型的方法步骤。该模型不仅有助于验证其设计, 还有利于实现实时任务接纳控制机制, 并为利用ASM模型方法设计实现其他控制机制提供了范例。

**关键词:** 抽象状态机; 共享资源网络; 实时任务; 可执行概率; 接纳控制

## Specifying Executable Probability Design of Real-Time Task in Resource-Sharing Network with ASM

ZHANG Dong-Song, JIN Shi-Yao, CHEN Fang-Yuan, LI Chao

(National Lab of Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Based on a formal method called Abstract State Machine (ASM), this paper describes an admission control mechanism of real-time task in resource-sharing network. In the meantime, the paper also proposes an ASM-based design model for executable probability of real-time task, and shows the methods and steps to analyze this design model based on Spec Explorer. The paper argues that the work will not only do good to further verification for the design method, but also to more refinements for admission control of real-time task. And the ASM model also provides an example for the realization of other control mechanisms designed.

**Key words:** abstract state machine; resource-sharing network; real-time task; executable probability; admission control

随着网络信息时代的到来, 大量的应用都需要网络进行收集、传输、处理和加工信息, 如视频播放、远程会议、网上教育、联合演习等等应用迅速发展。在这些应用中, 任务的执行往往具有一定的时限要求, 这使得网络环境中如何执行实时任务也成为人们关注的热点。但是现有网络在设计之初并没有考虑对实时任务执行的支持, 网络资源动态变化, 无法满足实时任务的时间要求, 不可以直接在当前网络上运行实时任务。如果重新设计符合实时性要求的网络, 包括新的网络结构、协议、软件等, 代价比较高, 很难实现。实时任务的时限要求与网络资源的随机变化成为一对突出的矛盾。因此, 在不改变现有网络结构的基础上, 利用共享网络资源执行实时任务, 给出实时任务可执

行概率, 进而实现实时接纳控制器, 这成为一种新的解决思路, 具有现实意义。

共享网络中实时任务可执行概率思想已经作为国防科技大学的国家发明专利《共享资源网络的实时任务接纳控制方法》(专利号 ZL200810144000.3) 而首次提出<sup>[1]</sup>, 但是深入的分析设计过程并没有给出, 不利于进一步的研究和验证。本文基于一种形式化方法: 抽象状态机 (Abstract State Machine, ASM) 方法<sup>[2]</sup>, 描述了共享网络中实时任务接纳控制机制, 提出并举例说明了实时任务可执行概率设计过程的ASM模型建立和应用。该模型不仅有助于验证其设计, 还为进一步实现实时任务接纳控制机制提供了可执行基础。

① 基金项目: 国家教育部博士点基金(20079998014, 20104307110005); 国防科学技术大学优秀研究生创新资助(B100601)

收稿时间: 2011-01-05; 收到修改稿时间: 2011-03-06

## 1 ASM简介

ASM 是 Dijkstra 抽象机 (Abstract machine) 与采用 Tarski 结构描述抽象状态 (Abstract state) 两个概念的结合, 于 1988 年由 Yuri Gurevich 首次提出<sup>[3]</sup>。Gurevich 在设计之初采用了动态结构进行抽象定义。随后, Borger 认识到动态结构在实践中的潜力, 用其解决了 Prolog 设计中的难题, 并定义了 Prolog 的动态语义标准<sup>[4]</sup>。此后, 人们认识到 ASM 适合于为动态系统建模。ASM 的严格数学定义首次形成于 1993 年, 开始被称为演化代数。1997 年后, ASM 的最终定义逐渐形成<sup>[5]</sup>。

目前, ASM 已经成功应用于设计包括虚拟机、微处理器体系结构、协议、嵌入式系统和大型软件需求等在内的众多领域。已出现许多关于 ASM 的开发工具, 例如 Spec Explorer<sup>[6]</sup>就是 Microsoft 开发的一种支持 .Net 的建模软件开发工具。

ASM 模型基本定义<sup>[5]</sup>包括:

定义 1. 抽象状态机=抽象状态 (Abstract State) + 抽象机器 (Abstract Machine)。

定义 2. 函数签名 (Signature) 由函数名称、关系名称及其固定的参数组成。规定 true, false, undef, =, 以及常规逻辑运算符均为函数签名。

定义 3. 状态 (State) 表示系统的一个瞬间配置, 是 Signature 上的一阶结构, 由一个非空集合和与其对应的函数签名组成。

定义 4. 域 (Universe) 是一元关系的特殊名称, 规定 undef 不是域。

定义 5. 项 (Term) 采用结构归纳法定义: 1) 一个变量是一个项; 2) 如果 f 是一个 n 元函数签名, 且  $t_1, \dots, t_n$  是项, 则  $f(t_1, \dots, t_n)$  也是项。

定义 6. 更新规则 (Update Rule) 表示为:  $f(t_1, \dots, t_n) := t$ 。

定义 7. 条件规则 (Conditional Rule) 表示为: if g then R1 else R2 endif。

定义 8. 并行规则 (Do-for-parallel Rule) 可表示为:

do in-parallel

R1

R2

Enddo

定义 9. 全称规则 (Do-for-all Rule) 可表示为:

do forall v:g(v)

R0(v)

Enddo

定义 10. 非确定规则 (Nondeterministic Rule) 通常表示为:

choose v:g(v)

R0(v)

endchoose

## 2 网络实时任务可执行概率设计

### 2.1 网络资源

本文研究的共享网络资源主要包括以下三类<sup>[1]</sup>:

1) 计算资源 (Computing Resource): 平均执行速度  $V_c$ , 计算极值  $C_{Net}$ ;

2) 存储资源 (Memory Resource): 存储速度  $V_m$ , 存储极值  $M_{Net}$ ;

3) 传输资源 (Traffic Resource): 流速  $V_t$ , 流量极值  $T_{Net}$ 。

对于任何网络而言, 每种资源都有利用率  $U$ , 不同资源有不同的额定值, 它们可使用值分别为  $U_c C_{Net}$ ,  $U_m M_{Net}$ ,  $U_t T_{Net}$ , 因此, 网络资源可形式化表述为:

Network Resource =  $\{(V_c, C_{Net}, U_c), (V_m, M_{Net}, U_m), (V_t, T_{Net}, U_t)\}$

### 2.2 网络实时任务可执行概率

简单的说, 网络实时任务可执行概率是指在用户的容忍范围内, 共享网络中的实时任务请求能够执行的概率是多少。这里所说的容忍范围主要是指如流媒体一类的弱硬实时任务<sup>[7]</sup>, 在执行过程中, 允许存在一定数量任务的超时, 超时后对系统性能损失影响不大。文献[1]不仅定义了网络实时任务可执行的充分必要条件, 还指出网络实时可执行概率的确定就是对网络实时任务可执行的必要和充分条件的判定。

#### 2.2.1 必要条件的判定

定理 1. 在整个实时任务 (集) 执行过程中, 任何时刻共享网络所能提供的资源余量必须满足实时任务 (集) 对资源的需求, 即实时任务 (集) 可执行的必要条件就是资源必须供大于等于求。

当在网络上执行新的实时任务  $T_i$  (或者实时任务集  $\{T_i\}$ ) 时, 它在执行过程中所需要的最大资源用量  $x_{T_{max}} (C_{T_{max}}, M_{T_{max}}, T_{T_{max}})$  都应该小于网络的资源余量  $x^*$ , 即有  $x^* \geq x_{T_{max}}$ , 满足实时必要条件的概率为:

$$p\{T_i\} = p\{x^* \geq x_{T_{max}}\} \geq \sum_{x^* \geq x_{T_{max}}} p(x^*)$$

如果共享资源的分布为连续函数时,

$$p\{T_i\} = \int_{x_{T \max}}^{x_{T \max}} p(x^*) dx^*$$

仅从必要条件来看, 如果网络实时任务在某一时段内可执行, 则它的可执行概率应为网络资源余量大于实时任务所需资源最大值的概率的总和。

### 2.2.2 充分条件的判定

定理 2. 实时任务的可执行充分条件就是任务的执行时间  $ET_i$  满足时限的要求。

已知在一组实时任务集中, 实时任务存在不同的周期, 记为  $(p_1, p_2, \dots, p_n)$ , 则计算不同任务之间的周期最小公倍数, 被称为超周期  $Lcm(p_1, p_2, \dots, p_n)$ 。由于满足网络实时任务可执行的必要条件并不保障实时任务的可执行, 网络的处理速度  $(V_c, V_m, V_i)$  可能达不到实时的时限要求, 所以还必须根据网络提供的处理速度, 来判断实时任务可执行的充分条件, 即是否满足:

$$\sum_{i=1}^n N_i \cdot ET_i \leq Lcm(p_1, p_2, \dots, p_n)$$

不能满足时, 可探讨增加  $(m, k)$  约束条件<sup>[8]</sup>。所谓  $(m, k)$  约束条件就是允许在连续执行  $k$  次实时任务中, 至多丢弃  $m$  次任务 (不执行或超时), 这样, 在  $(m, k)$  约束条件下的可执行充分条件即为

$$\sum_{i=1}^n N_i \frac{m_i}{k_i} ET_i \leq Lcm(p_1, p_2, \dots, p_n)$$

$m_i/k_i$  是约束系数  $\delta_i$ , 它是小于 1 的数。选取一定的  $\delta_i$  值, 则能确保上述不等式成立。

因此, 考虑充分条件, 在允许增加任意约束的条件下, 网络实时可执行的充分必要条件就是网络实时任务可执行概率。

### 2.3 实时任务接纳控制原理

文献[1]探索性地提出了实时任务接纳控制原理图, 如图 1 所示。该接纳控制是基于已成熟的网络流量控制接纳机制<sup>[9]</sup>, 并结合实时任务可执行概率的思想而发展出来的新兴机制。由于流量控制接纳机制仅需要对网络流量资源进行统计分析, 而实时任务接纳控制机制需要对网络资源进行统计分析, 所以相比单一的流量控制, 实时任务接纳控制更复杂。当新的实时任务进入网络边缘入口时, 需要先向接纳控制器发送探测包, 提出实时任务执行申请。而接纳控制器根据对当前网络资源状态的分析, 计算实时任务可执行

概率, 返回接纳决策通知。实时任务再根据决策通知送入网络执行, 最后形成输出结果。

显然, 接纳控制器是实时任务接纳控制的关键。它主要包括四个模块: 实时任务申请预处理模块、资源现状统计分析模块、实时任务判别模块和接纳决策模块。其中, 实时任务申请预处理模块主要通过探测包获取实时任务申请执行的各种需求。探测包处于任务包的头部, 具体包括了实时任务的执行时长即周期或超周期  $T_e$ , 绝对时限等属性, 以及所需各种资源的极大值, 额定最大容忍值  $(m, k)$ , 还有其他用户关心的事项等。资源现状统计分析模块则主要实现网络资源状态的统计、建模与分析。由于网络资源的时变性和不确定性, 在对新的任务请求做出接纳决策之前, 很难事先准确预知流量的模型, 因此本文没有采用基于模型的接纳控制方法要求<sup>[10]</sup>, 而基于测量的方法<sup>[11-13]</sup>对实时获取的网络资源状态进行分析, 较好地适应网络负载的动态变化, 形成网络资源余量的概率分布状态时效图。因篇幅所限, 相关成果请详见文献[14]。而实时任务判别模块和接纳决策模块, 分别实现了计算实时任务可执行概率与提供接纳决策的功能。

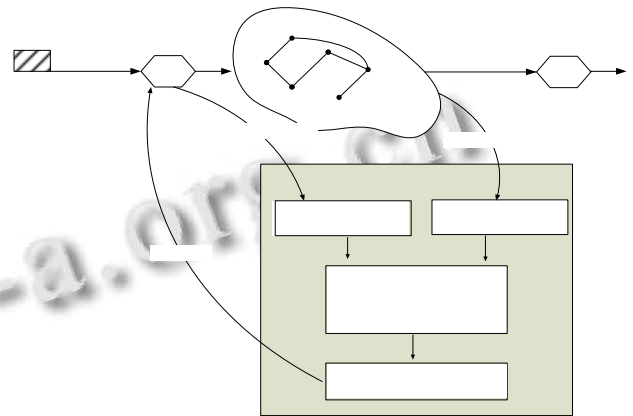


图 1 实时任务接纳控制原理图

### 2.4 计算实时任务可执行概率的设计

综上所述, 本文给出具体的计算实时任务可执行概率设计过程, 如图 2 所示, 首先根据实时任务探测包的内容, 形成执行时长  $T_e$  的固定窗口, 并在时效图上移动, 获得最优 (即资源余量超过需求量的概率最大) 的窗口时段, 再进行判断实时任务可执行的必要和充分条件。最后形成实时任务的可执行概率和接纳决策通知。

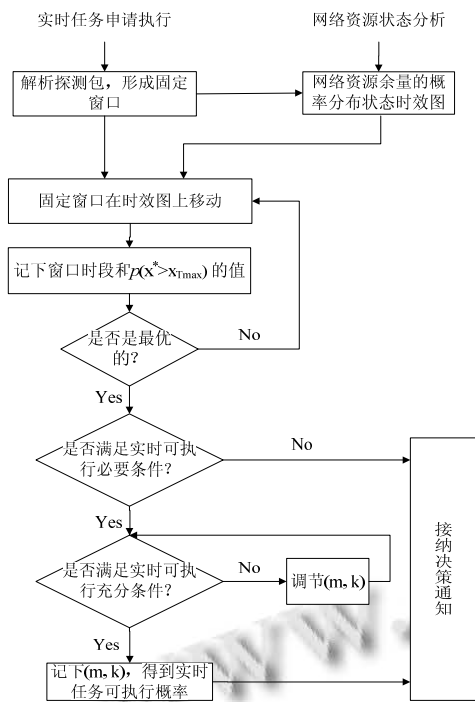


图 2 计算实时任务可执行概率的设计图

### 3 ASM模型建立

#### 3.1 ASM 模型中域、变量及函数的定义

实时任务可执行概率设计 ASM 模型的域定义如表 1 所示:

表 1 实时任务可执行概率设计 ASM 模型的域

域(Universe)	描述
BOOL	{true, false}
PROB	[0, 1]
TASK	{T <sub>1</sub> , T <sub>2</sub> , ..., T <sub>n</sub> }
JOB	{T <sub>ij</sub>   1 ≤ i ≤ n, 1 ≤ j ≤ Lcm(p <sub>1</sub> , p <sub>2</sub> , ..., p <sub>n</sub> )/p <sub>i</sub> }
JATTR	{a, r, c, p, d, m, k, C <sub>T</sub> , M <sub>T</sub> , T <sub>T</sub> }
RESOURCE	{computation, memory, traffic}
RATTR	{V <sub>c</sub> , C <sub>Net</sub> , U <sub>c</sub> , V <sub>m</sub> , M <sub>Net</sub> , U <sub>m</sub> , V <sub>t</sub> , T <sub>Net</sub> , U <sub>t</sub> }
TIMEWINDOW	(0, 24)
MESSAGE	全体消息

表 1 主要包括实时任务模型描述, 共享网络资源模型描述, 时效图中时间窗口描述, 以及概率描述。其中, BOOL 为布尔类型集合, 取值为真(true)或假(false)。PROB 既可表示实时任务的可执行概率, 又可表示资源余量超过需求量的概率。域 TASK 表示周期性实时任务, 每个周期任务又对应着多个实例域 JOB。

如果实时任务集表示为{T<sub>1</sub>, T<sub>2</sub>, ..., T<sub>n</sub>}, 则每个任务

的一次实例 JOB = {T<sub>ij</sub> | 1 ≤ i ≤ n, 1 ≤ j ≤ Lcm(p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>)/p<sub>i</sub>}, Lcm(p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>)表示所有实时任务的超周期, i 表示实时任务的序号, 而 j 对应着该实时任务的实例序号。为了较详细描述每个任务的属性, 任务属性域 JATTR(T<sub>ij</sub>)={ (a<sub>ij</sub>, r<sub>ij</sub>, c<sub>ij</sub>, p<sub>ij</sub>, d<sub>ij</sub>, m<sub>i</sub>, k<sub>i</sub>, C<sub>Ti</sub>, M<sub>Ti</sub>, T<sub>Ti</sub>) }, 其中 a<sub>ij</sub> 是任务 T<sub>ij</sub> 的到达时间, r<sub>ij</sub> 是释放时间, c<sub>ij</sub> 是最坏情况下执行时间, p<sub>ij</sub> 是周期, d<sub>ij</sub> 是绝对时限, (m<sub>i</sub>, k<sub>i</sub>) 则说明该任务容忍丢失时限的最大值, 而分别 C<sub>Ti</sub>, M<sub>Ti</sub>, T<sub>Ti</sub> 是任务 T<sub>ij</sub> 所请求的资源量。

域 RESOURCE 定义了共享网络的各种资源, 主要包括计算资源、存储资源和传输资源。资源属性域 RATTR 可以表示为{(V<sub>c</sub>, C<sub>Net</sub>, U<sub>c</sub>), (V<sub>m</sub>, M<sub>Net</sub>, U<sub>m</sub>), (V<sub>t</sub>, T<sub>Net</sub>, U<sub>t</sub>)}, 其中 V<sub>c</sub> 是平均执行速度, C<sub>Net</sub> 是计算极值, U<sub>c</sub> 是计算利用率, V<sub>m</sub> 是存储速度, M<sub>Net</sub> 是存储极值, U<sub>m</sub> 是存储利用率, V<sub>t</sub> 是流速, T<sub>Net</sub> 是流量极值, U<sub>t</sub> 是传输利用率。

域 TIMEWINDOW 定义了时效图中固定时间窗口的大小。由于时效图表示一天 24 小时内每个时段的资源余量变化情况, 所以设置时间窗口的大小不超过一天。

实时任务可执行概率设计中需要定义的变量如表 2 所示:

表 2 实时任务可执行概率设计中变量定义

变量	描述
T <sub>i</sub> : TASK	当前执行的任务
J <sub>i</sub> : JOB	每个任务 T <sub>i</sub> 的实例
pr(1): RESOURCE	当前剩余网络计算总资源
pr(2): RESOURCE	当前剩余网络存储总资源
pr(3): RESOURCE	当前剩余网络传输总资源
ar(1): RESOURCE	实时任务请求计算总资源
ar(2): RESOURCE	实时任务请求存储总资源
ar(3): RESOURCE	实时任务请求传输总资源
Jpmax(J): PROB	任务实例可执行概率
RProb(pr): PROB	资源余量超过需求量的概率
tsize: TIMEWINDOW	标记固定时间窗口的大小
time_s(J): TIMEWINDOW	任务实例最优窗口的起始时间

实时任务可执行概率设计中的函数签名应包括:

- jattr: {JOB, TASK}--> JATTR (获得任务属性)
- rattr: RESOURCE--> RATTR (获得资源属性)
- necessity: RATTR × RATTR--> {true, false} (获得实时任务可执行的必要性)

● **efficient**:  $JATTR \times TIMEWINDOW \rightarrow \{true, false\}$  (获得实时任务可执行的充分性)

● **mappedResource**:  $JOB \times RESOURCE \rightarrow RESOURCE$  (将任务实例和请求资源映射为网络剩余资源)

● **state**:  $JOB \rightarrow \{arrival, receive-waiting, waiting, running, missing\}$

● **request**:  $JOB \times RESOURCE \rightarrow \{true, false\}$  (任务实例请求资源)

● **task**:  $JOB \rightarrow TASK$  (获得任务实例有关任务)

● **type**:  $RESOURCE \rightarrow \{computation, memory, traffic\}$  (获得资源类型)

● **Lcm**:  $R \times R \rightarrow R$  (求两个实数的最小公倍数, 其中  $R$  表示实数)

● **ProbGraph: RATTR  $\times$  RATTR  $\rightarrow$  TIMEWINDOW  $\times$  PROB** (获得超过实时任务资源请求的概率时效图)

● **requestProbGraph: TIMEWINDOW  $\times$  PROB  $\rightarrow$   $\{true, false\}$**  (请求概率时效图)

● **timeJProb: TIMEWINDOW  $\rightarrow$  PROB** (获得固定窗口在时效图上的概率值)

● **reductimeJProb: TIMEWINDOW  $\times$  TIMEWINDOW  $\times$  PROB  $\rightarrow$  TIMEWINDOW  $\times$  PROB** (获得分配资源后的概率时效图)

● **scaleProb: JATTR  $\times$  PROB  $\rightarrow$   $\{true, false\}$**  (根据充分条件不满足的判断调节概率值)

● **notice: PROB  $\times$  TIMEWINDOW  $\times$  JOB  $\rightarrow$  MESSAGE** (实时任务实例的接纳决策通知)

● **to: MESSAGE  $\rightarrow$  JOB** (发送消息)

● **from: MESSAGE  $\rightarrow$  JOB** (接受消息)

### 3.2 ASM 模型中初始条件和规则的定义

采用 ASM 描述可执行概率设计的初始条件为:

1)  $\forall J_1, J_2, \dots, J_n \in JOB, 1 < i < n, \exists r \in RESOURCE:$

$request(J_i, r) = true$

2)  $\forall J_i \in JOB, 1 < i < n: state(J_i) = arrival$

3)  $\exists ar, pr \in RESOURCE: tsize = 1 \wedge time\_s = 0 \wedge$

$Newar = undef \wedge Newpr = undef \wedge msg = undef$

4)  $\forall J_i \in JOB, 1 < i < n, \exists pr \in RESOURCE:$

$Jpmax(J_i) = 1 \wedge RProb(pr) = 0$

采用 ASM 描述可执行概率设计的更新规则包括:

● 规则 1: 解析探测包

if  $\forall J \in JOB \wedge \forall msg \in MESSAGE \wedge \exists r \in RESOURCE: from(msg) = J \wedge state(J) = arrival \wedge request(J, r) = true$  then

do forall  $T_i \in TASK: task(J) = T_i$

$tsize = Lcm(tsize, T_i.p)$

enddo

$rattr(ar(1)).C_{Net} = rattr(ar(1)).C_{Net} + jattr(J).C_{ti}$

$rattr(ar(2)).M_{Net} = rattr(ar(2)).M_{Net} + jattr(J).M_{ti}$

$rattr(ar(3)).T_{Net} = rattr(ar(3)).T_{Net} + jattr(J).T_{ti}$

$state(J) = receive-waiting$

endif

● 规则 2: 形成状态时效图

if  $\forall J \in JOB \wedge \exists r \in RESOURCE: state(J) = receive-waiting \wedge r' \mapsto mappedResource(J, r) = undef$  then

if  $rattr(pr(1)).C_{Net} \neq undef$  then ProbGraph( $rattr(pr(1)).C_{Net}, rattr(ar(1)).C_{Net}$ ) endif

if  $rattr(pr(2)).M_{Net} \neq undef$  then ProbGraph( $rattr(pr(2)).M_{Net}, rattr(ar(2)).M_{Net}$ ) endif

if  $rattr(pr(3)).T_{Net} \neq undef$  then ProbGraph( $rattr(pr(3)).T_{Net}, rattr(ar(3)).T_{Net}$ ) endif

endif

● 规则 3: 移动固定窗口获得每个任务实例的最优窗口时段和概率值

if  $\forall J \in JOB \wedge \forall t \in TIMEWINDOW \wedge \exists p \in PROB: requestProbGraph(t, p) = true \wedge jpmax(J) = undef$  then

do forall time\_s:  $time\_s \leq 24 - tsize$

if  $timeJProb(time\_s) > jpmax(J)$  then  $jpmax(J) = timeJProb(time\_s)$  endif

$time\_s(J) = time\_s(J) + tsize$

enddo

$reductimeJProb(time\_s(J), time\_s(J) + tsize, jpmax(J))$

(J)

endif

● 规则 4: 判断可执行必要条件是否满足

if  $\forall J \in JOB \wedge \exists pr \in RESOURCE \wedge \exists ar \in RESOURCE: time\_s(J) \neq undef \wedge jpmax(J) \neq undef \wedge necessity(rattr(pr), rattr(ar)) = false$  then  $jpmax(J) = 0$

$time\_s(J) = undef$

$to(notice(jpmax(J), time\_s(J), J)) = J$

$state(J) = missing$

endif

● 规则 5: 判断实时可执行充分条件是否满足

```

if  $\forall J \in \text{JOB} \wedge \exists pr \in \text{RESOURCE} \wedge \exists ar \in \text{RESOURCE}: \text{time\_s}(J) \neq \text{undef} \wedge \text{jpmax}(J) \neq \text{undef} \wedge \text{necessity}(\text{rattr}(pr), \text{rattr}(ar)) = \text{true}$  then
  if  $\text{efficient}(\text{jattr}(J), \text{tsize}) = \text{true}$  then
    if  $\text{jpmax}(J) > (\text{jattr}(J).m / \text{jattr}(J).k)$  then to
      ( $\text{notice}(\text{jpmax}(J), \text{time\_s}(J), J)$ ):=J
    else to( $\text{notice}((\text{jattr}(J).m / \text{jattr}(J).k), \text{time\_s}(J), J)$ ):=J
  endif
else
  scaleProb( $\text{jattr}(J)$ ,  $\text{jpmax}(J)$ )
endif
endif

```

#### 4 ASM模型应用

ASM 模型是对共享资源网络中实时任务可执行概率设计系统的一种数学模型描述, 如何应用是一个相对复杂的过程。Spec Explorer 是 Microsoft 公司针对 ASM 模型开发的一种最新软件开发工具, 为 ASM 模型描述和测试提供了支持。借助 Spec Explorer, ASM 模型的应用通常需要完成以下三个步骤<sup>[3]</sup>: 编程、测试和比较。编程是指为系统行为的描述生成 ASM 模型程序。通常, 模型程序要比系统实现简单的多, 但它足以说明系统相关的状态以及为正确实现而应遵循的约束。测试是指描述模型程序所有可能的运行, 用以验证系统设计的有效性。比较是指在系统实现中, 将实际场景中系统行为与模型程序中对应的行为进行对比分析。

ASM 模型具有精确性和自然性, 可以形式化描述共享网络中实时任务的接纳控制机制。用户在 Spec Explorer 中按照 ASM 模型定义接纳控制机制的规则和行为, 验证该接纳控制系统的有效性。此时用户可以开始进行测试, 设置初始状态以及域取值, 以验证设计的合理性。图 3 左侧文档窗口显示所编写的 ASM 模型程序, 而模型的初始状态和每个域的取值范围可在右侧列表查看。编译通过后运行可以得到相应的有限状态机 (Finite State Machine, FSM), 如图 4 左侧窗口所示。通过有限状态机用户可以形象直观地从逻辑角度检测软件的内部一致性, 验证软件设计的合理性。

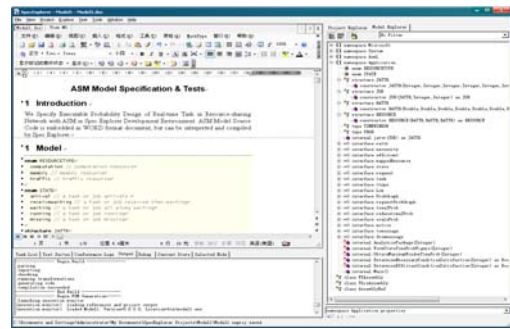


图 3 在 Spec Explorer 中 ASM 模型程序

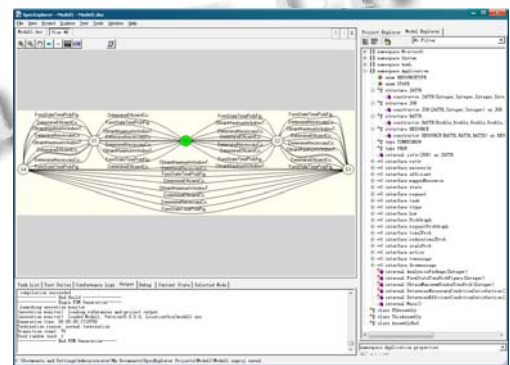


图 4 ASM 模型程序的有限状态机

#### 5 结语

共享资源网络中实时任务可执行概率设计是在不改变现有网络的条件下实现实时任务接纳控制的关键。本文基于抽象状态机的形式化方法, 针对共享资源网络中实时任务可执行概率设计思路, 建立了 ASM 模型, 通过实例说明了 ASM 模型的应用。本文提出的 ASM 模型不仅有助于验证实时任务可执行概率设计, 还有利于实现实时任务接纳控制机制, 并为利用 ASM 模型方法设计实现其他控制机制提供了范例。

#### 参考文献

- 1 金士尧, 等. 共享资源网络的实时任务接纳控制方法. 中国, 专利号: ZL200810144000.3, 2010-12-01.
- 2 Gurevich Y. Sequential abstract state machines capture sequential algorithms. *ACM Trans. on Computational Logic*, 2000, 1(1): 77-111.
- 3 Borger E. The origins and development of the ASM method for high level system design and analysis. *Journal of Universal Computer Science*, 2002, 8(1): 2-74.
- 4 Abstract State Machines. <http://www.eecs.umich.edu/gasm>

(下转第 16 页)

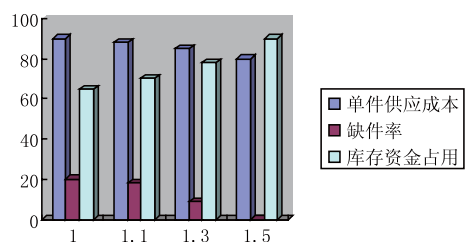


图4 系统仿真运算结果分析

### 3 结语

装备制造企业生产过程是一类典型的复杂系统,采用传统的研究方法很难描述和求解其多阶段、多层次和多主体间的动态关系,企望通过复杂的数学建模过程研究该类物流链问题并不切合实际应用;基于计算机的建模仿真方法是研究物流链的最有力的工具之一。文章针对装备制造企业生产物流的特点,提出了基于离散事件建模的生产物流过程模型,设计了装备制造企业生产物流仿真系统,以某大型企业为研究背景,量化分析了生产过程物流的关键因素。

鉴于装备制造企业生产过程物流的复杂性,本文提出的仿真系统仍需进一步完善,尤其是关键仿真参数的获取,对于仿效运行效果至关重要,如何利用企业积累的历史数据对仿真参数进行修正是未来的研究重点。

### 参考文献

- 1 刘玉坤,石伟,张晓萍. 集合标志 Petri 网的建模方法及在供应链库存控制中的应用. 计算机集成制造系统, 2005, 11(9): 1260-1266.
- 2 Sanjay J, Workman RW, Collins LM, et al. Development of a high-level supply chain simulation model. [2003-01-10]. <http://www.informs-cs.org/wsc01papers/153.pdf>
- 3 姜金菊,林杰. 基于智能代理的供应链仿真. 系统仿真学报, 2004, 12: 2847-2850.
- 4 王忠明. 基于过程集成的工装业务流程管理研究[硕士学位论文]. 西安: 西北工业大学, 2005.
- 5 郑锋. 混合型生产过程建模与调度优化[博士学位论文]. 西安: 西北工业大学, 2003.
- 6 Yun WY, Choi YS. A simulation model for container terminal operation analysis using an object-oriented approach. *Production Economics*, 1999, 59(123): 221-230.
- 7 张新艳,周健. DFT 的装配车间生产物流策略仿真与研究. *工业工程*, 2007, 9: 114-118.
- 8 曹渝昆,王海林. 基于状态检修的流域性水电厂备品配件系统. *电网技术*, 2007, S2: 162-165.
- 9 杨宇航,赵建民,李志忠,郑力. 备品配件管理系统仿真研究. *系统仿真学报*, 2004, 16(5): 981-984.
- 10 张涛,苗明,金淳. 基于仿真优化的集装箱堆场资源配置研究. *系统仿真学报*, 2007, 19(24): 5631-5634.

(上接第 53 页)

- 5 刘晖,李明禄. 基于抽象状态机的网格系统设计和分析. *电子学报*, 2003, 31(12A): 2096-2100.
- 6 Spec Explorer. <http://research.microsoft.com/en-us/projects/specexplorer/>
- 7 Guillem B, Alan B, Albert L. Weakly hard real-time systems. *IEEE Trans. on Computers*, 2001, 50(4): 308-321.
- 8 Hamdaoui M, Ramanathan P. A dynamic priority assignment technique for streams with (m,k)-firm deadlines. *IEEE Trans. on Computer*, 1995, 44(12): 1443-1451.
- 9 张光胜. MPLS 接纳控制关键技术研究. 长沙: 国防科技大学, 2008.
- 10 Lee T K, Zukerman M. Practical approaches for connection admission control in multiservice networks. *Proc. IEEE ICoN'99*. 1999: 172-177.
- 11 Jamin S, Shenker SJ, Danzig PB. Comparison of measurement-based admission control algorithms for controlled-load service. *IEEE INFOCOM*, 1997: 973-980.
- 12 TSE D, Orossglauser M. Measurement-based call admission on control: Analysis and simulation. *IEEE INFOCOM*, 1997: 981-989.
- 13 马小骏,顾冠群. 基于测量的接纳控制研究. *计算机学报*, 2001, 24(1): 40-45.
- 14 李超,张冬松,金士尧. 面向流媒体应用的实时可执行概率获取方法研究. *计算机应用研究*, 2010, 27(9): 3462-3465.