

基于面向方面的实时系统建模方法^{*}

刘瑞成 张立臣

(广东工业大学计算机学院 广州 510090)

摘要 分布式实时系统的实时特性可以利用面向方面软件设计方法来建模,把时间方面细分为确定的时间子方面、不确定的时间子方面和模糊时间子方面。根据面向方面技术,不同的时间子方面分别利用随机实时时序逻辑(SQTL)和模糊时间 Petri 网(FTN)来表示,并且每个不同形式化语言表示的子方面模型能够通过转化为时间自动机织入系统,实现系统的实时特性。

关键词 面向方面软件设计,形式化方法,时间自动机,模糊时间 Petri 网,随机实时时序逻辑,实时系统

Real-time System Modeling Method Based on Aspect-oriented

LIU Rui-Cheng ZHANG Li-Chen

(Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510090)

Abstract The real-time of distributed real-time systems can be modeled by aspect-oriented software design methods, and separated into deterministic timing sub-aspect, undeterministic timing sub-aspect and fuzzy timing sub-aspect further. According to the aspect-oriented technology, different timing sub-aspects can be designed by different design languages, such as the Stochastic and Real-time Temporal Logic for deterministic and underministic timing sub-aspects, and Fuzzy-timing Petri Nets for the fuzzy timing sub-aspect. After all the timing aspects are designed, they are translated into timed automata and woven into the system, and realize the real-time features.

Keywords Aspect-oriented software design, Formal method, Timed automata, Fuzzy-timing Petri nets, Stochastic and real-time temporal logic, Real-time system

1 引言

面向方面软件设计(AOSD)方法是基于关注分离技术,能够把系统划分为不同的方面,并且每个方面能够单独地设计和实现。面向方面编程(AOP)^[1]技术允许系统不同的方面利用不同的语言设计和实现,设计和实现之后的所有方面能够重新织入,组合为一个系统。AOSD 在不需要改动原有模型的情况下,能动态地修改原有模型。

实时系统的面向方面建模就在于能把时间方面独立处理,因此可以建立一个时间方面模型来统一表达系统的时间特性。并且,这个时间模型可以根据需要进一步细化为多个子方面,例如确定时间方面、随机时间方面和模糊时间方面等,每个时间子方面根据各自的特点利用不同的形式化方法来表示。这些不同的时间子方面实现之后能够重新构造造成一个时间方面,并管理着系统的实时性。

分布式实时系统的时间分析是非常复杂的,不同的分布式实时系统具有不同的时间特性需求。它的时间方面包括确定的时间、不确定的时间和模糊时间等。为了简化多种时间混淆在一起形成的复杂性,我们可以把这些时间进行分类,并分离出来形成子时间方面。根据面向方面的思想,不同的时间子方面还可以利用不同的形式化语言来表达^[2]。在实现所有方面模型之后,又可以把它们织入到系统,完成它在系统特定的作用。目前,各种形式化语言都是通过扩展实时性来建模实时系统,还没有一种较完善的语言能够表达所有的时间特性。Lakas&Blair 提出的随机实时时序逻辑(SQTL)^[3]能

够很好地表达确定的时间和不确定的时间,并且可以转化为时间自动机,方便时间子方面的织入。因此,我们利用 SQTL 来表达确定的时间子方面和不确定的子方面。而模糊子方面我们采用模糊时间 Petri 网(FTN)^[4~6]来建模。

2 随机实时时序逻辑及其转化方法

定义 1 SQTL 公式可以定义如下:

$$\Psi := a \mid \text{Ture} \mid \neg\psi \mid \psi \vee \psi \mid \psi \dot{\vee} \psi \mid O_{\tau} \psi \mid \Theta_{\tau} \psi \mid \psi U_{\tau} \psi \mid \psi S_{\tau} \psi$$

$a \in A$, A 为原子动作集;

$p \in [0, 1]$ 是事件发生概率;

Q 表示任意常数或指数分布;

$< \in \{=, <, \leq\}$;

$O\psi$ 表示在下一个状态公式 ψ 仍然为真;

$\Theta\psi$ 表示前一个状态公式 ψ 为真;

$\psi_1 U \psi_2$ 表示公式 ψ_1 从当前状态开始的所有状态都为真,直到将来某个状态使公式 ψ_2 为真;

$\psi_1 S \psi_2$ 表示公式 ψ_2 在过去某个状态为真,且从此状态到当前状态公式 ψ_1 总为真。

另外,可以从上面的基本操作中导出其它的时序操作,例如 $\diamond p \equiv \text{True} U p$, $\square p \equiv \neg \diamond \neg p$ 。 $\diamond\psi$ 表示公式 ψ 在将来的某个状态必定为真, $\square\psi$ 表示公式 ψ 在将来的所有状态都为真; $\triangleleft\psi$ 表示公式 ψ 在过去的某个状态为真, $\triangleleft\psi$ 表示公式 ψ 在过去的所有状态都为真。

假设无限的带时间状态序列为 $\sigma_i(s_0, \tau_0), (s_1, \tau_1), \dots, (s_i, \tau_i), \dots$, 其中 s_i 为状态, τ_i 为对应状态的时间值,时间间隔为

^{*} 本文受国家自然科学基金(No. 60474072、No. 60174050)、广东省自然科学基金(No. 04009465、No. 010059)、广东省高校自然科学研究项目(No. Z03024)基金资助。刘瑞成 硕士研究生,主要研究方向:实时系统、软件设计方法与实时软件设计方法;张立臣 博士,教授,硕士生导师,主要研究方向:并行处理、分布式处理、网络计算、实时系统等。

$I, I_i = \tau_{i+1} - \tau_i; D$ 为时间延迟域, $D_{x,\lambda} \subseteq D$, 表示以 λ 为参数的 x 分布产生的时间间隔; 则时序公式 ψ 满足下列关系:

- $(\sigma, \tau_i) \models \text{True}$;
- $(\sigma, \tau_i) \models a$, 当 $a \in I(\sigma, \tau_i)$ 时;
- $(\sigma, \tau_i) \models \neg\psi$, 当 $(\sigma, \tau_i) \not\models \psi$ 时;
- $(\sigma, \tau_i) \models \psi_1 \vee \psi_2$, 当 $(\sigma, \tau_i) \models \psi_1$ 或 $(\sigma, \tau_i) \models \psi_2$ 时;
- $(\sigma, \tau_i) \models \psi_1 \dot{\vee} \psi_2$, 当以概率 p 满足 $(\sigma, \tau_i) \models \psi_1$ 且概率 $1-p$ 满足 $(\sigma, \tau_i) \models \psi_2$;
- $(\sigma, \tau_i) \models O_{x,\lambda}\psi$, 当 $\exists I \in D_{x,\lambda}, (\sigma, \tau_{i+1}) \models \psi$ 且 $\tau_{i+1} = \tau_i + I$;
- $(\sigma, \tau_i) \models \psi_1 U_{x,\lambda}\psi_2$, 当 $\exists I \in D_x, \lambda$ 有 $(\sigma, \tau_i + I) \models \psi_2$ 且 $\forall I' \in D$ 有 $0 \leq I' < I: (\sigma, \tau_i + I') \models \psi_1$ 。

3 模糊时间 Petri 网及其转化方法

定义 2 模糊时间 Petri 网定义为一个 7 元组 (P, T, A, M, D, FT, CT) , 其中

- $P = \{p_1, p_2, \dots, p_m\}$ 是库所集;
- $T = \{t_1, t_2, \dots, t_n\}$ 是迁移集;
- $A \subseteq (P \times T) \cup (T \times P)$ 表示边的集合;
- $M: P \rightarrow I, I = \{0, 1, 2, \dots\}$, 表示给每个库所赋予标志;
- D 是所有关联到边 $\text{arcs}(t, p) \subseteq (T \times P)$ 的模糊延迟 $d_p(\tau)$ 的集合, 表示从迁移 t 到库所 p 所需的延迟;

FT 是所有模糊时间戳的集合; 模糊时间戳 $\pi(\tau)$ 表示令牌在 τ 时刻到达特定库所的可能性, 它可能是一个模糊时间函数或可能性分布函数;

$CT: T \rightarrow [\alpha, \beta]$, 表示从迁移到点燃的时间间隔。

另外, 一个迁移 t 的所有对应令牌都到达相应库所的时间称为模糊使能时间 $e_t(\tau)$, 其计算公式为: $e_t(\tau) = \text{latest}\{\pi_i(\tau), i=1, 2, \dots, n\}$, 其中 latest 表示 n 个令牌中最慢到达或最小可能性分布时间, $\pi_i(\tau)$ 为令牌到达迁移 t 第 i 个输入库所的模糊时间戳。迁移 t 发生的时间为模糊发生时间 $o_t(\tau)$, 可以通过下列公式计算: $o_t(\tau) = \min\{e_t(\tau) \oplus p_i(\alpha_i, \alpha_i, \beta_i, \beta_i), \text{earliest}\{e_i(\tau) \oplus p_i(\alpha_i, \alpha_i, \beta_i, \beta_i), i=1, 2, \dots, l, \dots, m\}\}$, 其中 earliest 表示 m 个分布中最早到达或最大可能性分布时间, \min 表示对应分布的交集。模糊时间戳、模糊发生时间和模糊延迟之间的关系如下:

$$\pi_p(\tau) = o_t(\tau) \oplus d_{tp}(\tau)$$

4 分布式实时系统的时间方面

分布式实时系统的时间可以分为确定的时间、不确定的时间和模糊时间等。我们利用 SCTL^[3] 来表达确定的时间子方面和不确定的时间子方面。而模糊时间子方面我们采用模糊时间 Petri 网 (FTN)^[4,5,7] 来建模。分布式多媒体具有很强的实时性, 是一种典型的分布式实时系统。因此, 下面就结合分布式多媒体来说明。

4.1 确定的时间子方面

我们把能够事先确定的时间点归结为确定的时间, 并利用随机实时时序逻辑来表达。例如多媒体服务器数据源每隔 50ms 发送一帧数据, 当客户端收到数据延迟 5ms 立即播放。

$$(\text{source-out} \rightarrow \bigvee_{i=0}^{50} O_{i,5} (\neg(\text{source-out}) \cup_{i=50-i} \text{source-out}))$$

$$(\text{rec} \rightarrow \diamond_{=5} \text{play})$$

4.2 不确定的时间子方面

当系统的时间是一个范围, 而没有确切的值时, 我们归结为不确定的时间。例如假设数据传输通道端到端延迟必须在

80ms 到 90ms 之间, 否则视为丢失帧。数据丢失率为 0.1, 丢失的数据经过一段指数分布函数为 $X(I) = 0.2e^{-0.2I}$ 的随机时间延迟后进行重发。

$$(\text{send} \rightarrow (O(\text{loss} \rightarrow \diamond_{=e,0.2} \text{send}) \vee \diamond_{\leq 10\text{ms}} \text{rec}))$$

4.3 模糊时间子方面

在分布式实时系统中, 还有一种时间发生事件, 其发生不但在一段时间内不确定, 而且在这段时间内不同的时间点事件发生的可能性也不一样。这些, 我们归结为模糊时间。

例如假设数据在准备好发送之后的 2ms 到 4ms 之间必须发送出去, 当数据到达客户端时, 在之后的时间段 [2, 4]ms 内必须接收, 传输通道端到端延迟服从如图 1 所示的矩形分布, 则用模糊 Petri 网表示如图 2 所示。

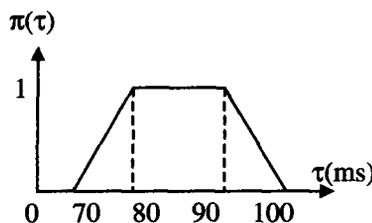


图 1 模糊时间分布

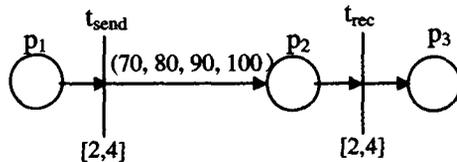


图 2 模糊 Petri 网表示

5 方面的织入

面向方面软件设计方法在设计好各个方面之后, 必须对方面进行“织入”, 即实现各个方面的系统组合。由于我们的设计是基于不同形式化语言来实现的, 不能直接进行不同语言的方面织入。因此, 我们先把各种实现方面的语言转化为时间自动机, 然后再进行织入系统。这里主要考虑各个时间方面与系统行为的同步问题, 系统的行为根据不同的环境被不同的时间方面所监测和同步。

5.1 随机实时时序逻辑的转换方法

随机实时时序逻辑转换为时间自动机可以通过定义一系列推理规则, 然后把 SCTL 公式转化为时间自动机。推理规则^[3]如下所示:

$$D_a[\text{True}] = \text{True}$$

$$D_a[a] = \text{True 当 } a = \alpha \text{ 时, 否则为 False}$$

$$D_a[\neg\psi] = \neg D_{a,\alpha}[\psi]$$

$$D_a[\psi_1 \vee \psi_2] = D_a[\psi_1] \vee D_a[\psi_2]$$

$$D_{a,T=\delta}[O_{=\delta}\psi] = \psi, \text{ 并初始化 } T \leftarrow 0$$

$$D_a[\psi_1 U_{=\delta}\psi_2] = D_{a,T=\delta}[\psi_2] \vee [D_{a,T \leq \delta}[\psi_1] \wedge (\psi_1 U_{=\tau}\psi_2)],$$

并初始化 $T \leftarrow 0$

其它的时序操作推理规则可以根据 SCTL 性质从上面基本规则中推导出, 转化为时间自动机的过程就是对公式的推导过程。例如对应公式 $\psi = (\alpha \rightarrow \diamond_{\leq \text{exp},5} b)$, 它的推导过程如下:

$$D_a[\psi] = D_a[\alpha \rightarrow \diamond_{\leq \text{exp},5} b] \wedge \psi = (D_a[\alpha] \rightarrow D_a[\diamond_{\leq \text{exp},5} b]) \wedge \psi = (\text{true} \rightarrow (D_a[b] \vee \diamond_{\leq \text{exp},5} b)) \wedge \psi = \diamond_{\leq \text{exp},5} b \wedge \psi$$

$$\begin{aligned}
 D_b[\psi] &= D_b[a \rightarrow \diamond_{\leq \text{exp},5} b] \wedge \psi = (D_b[a] \rightarrow D_b[\diamond_{\leq \text{exp},5} b]) \wedge \psi = (\text{false} \rightarrow (D_b[b] \vee \diamond_{\leq \text{exp},5} b)) \wedge \psi = \psi \\
 D. [\psi] &= D. [a \rightarrow \diamond_{\leq \text{exp},5} b] \wedge \psi = (D. [a] \rightarrow D. [\diamond_{\leq \text{exp},5} b]) \wedge \psi = (\text{false} \rightarrow (D. [b] \vee \diamond_{\leq \text{exp},5} b)) \wedge \psi = \psi \\
 \text{设 } \psi' &= (a \rightarrow \diamond_{\leq \text{exp},5} b) \wedge \diamond_{\leq \text{exp},5} b, \text{ 则} \\
 D_a[\psi'] &= D_a[\psi] \wedge D_a[\diamond_{\leq \text{exp},5} b] = (\diamond_{\leq \text{exp},5} b \wedge \psi) \wedge (D_a[b] \vee \diamond_{\leq \text{exp},5} b) = (\diamond_{\leq \text{exp},5} b \wedge \psi) \wedge \diamond_{\leq \text{exp},5} b = \psi' \\
 D_b[\psi'] &= D_b[\psi] \wedge D_b[\diamond_{\leq \text{exp},5} b] = \psi \wedge (D_b[b] \vee \diamond_{\leq \text{exp},5} b) = \psi \wedge \text{true} = \psi \\
 D. [\psi'] &= D. [\psi] \wedge D. [\diamond_{\leq \text{exp},5} b] = \psi \wedge (D. [b] \vee \diamond_{\leq \text{exp},5} b) = \psi \wedge \diamond_{\leq \text{exp},5} b = \psi'
 \end{aligned}$$

从上面的推导过程可以看出,这个公式可以转化为包含两个状态的时间自动机。如果 s_0 表示初始状态, s_1 表示另一状态,则转化后的状态机如图 3 所示。

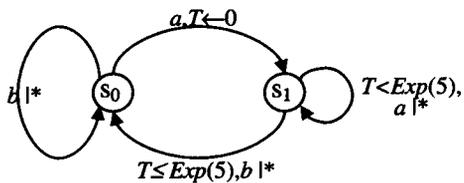


图 3 公式 $\psi = (a \rightarrow \diamond_{\leq \text{exp},5} b)$ 对应的的时间自动机

5.2 模糊时间 Petri 网的转换方法

我们通过扩展文[7]的方法来转化模糊时间 Petri 网。首先,定义模糊时间集为 $ft(\tau)$,并令 $\pi_i \in ft(\tau), e_i \in ft(\tau), o_i \in ft(\tau)$ and $d_i \in ft(\tau)$ 。两个模糊时间 α 和 β 的关系定义为 $later(\alpha, \beta)$,表示为 $\alpha \ll \beta$,其中 $later$ 是两个分布中较迟到达或较小可能性分布操作。因此,我们定义的模糊时间 Petri 网对应的的时间自动机如下:

定义 3 模糊时间自动机定义为一个 3 元组 $FTA = (S, s_0, \rightarrow)$,其中

$$\begin{aligned}
 S &= M \times ft(\tau); \\
 s_0 &= (M_0, ft(0)); \\
 \rightarrow &\in S \times (T \cup ft(\tau)) \times S, \text{ 并由下列关系组成: } \forall t \in T \\
 (M, v) &\xrightarrow{t} (M', v') \text{ iff } \begin{cases} M \geq Pre_t \wedge M' = M - Pre_t + Post_t \\ e_i(\tau) \circ \cup \\ v' = \begin{cases} 0 & \text{如果 } O_i(\tau) \circ \cup \\ \pi_i(\tau) = O_i(\tau) \oplus d_{ip}(\tau) & \text{其它} \end{cases} \end{cases}
 \end{aligned}$$

具体的转化步骤大概如下:

- 1) 为 FTN 的每个库所定义一个全局变量 p_i ;
- 2) 假设迁移 t 到点燃的时间间隔为 $[\alpha, \beta]$, 并且有 k 个输入库所分别为 p_1, \dots, p_k, l 个输出库所分别为 p_m, \dots, p_n ; 迁移 t 点燃后输入库所消耗的令牌分别为 pre_1, \dots, pre_k , 输出库所得到的令牌分别为 $post_m, \dots, post_n$; 那么在自动机中定义 p_i 为相应库所令牌数的变量, 而 $pre_i, post_i$ 为相应消耗和得到的令牌数。

3) 对于每个迁移定义一个具有 3 个状态的模糊时间自动机, 状态分别为 dis_i, enb_i 和 $firing_i$, 时钟为 c , 当迁移对应的所有库所中第一个令牌到达时开始计时。

4) 增加从 enb_i 状态开始的边, $\langle enb_i, firing_i \rangle$ 对应条件为 $p_i := p_i - pre_i, \dots, p_k := p_k - pre_k; \alpha(t) \leq c \leq \beta(t)$, 其中 $\alpha(\tau) = \alpha + \tau, \beta(\tau) = \beta + \tau, \tau$ 为使能时间; $\langle enb_i, dis_i \rangle$, 对应条件为 $p_i < pre_i; \langle enb_i, enb_i \rangle$, 对应条件为 $p_i \geq pre_i, \dots, p_k \geq pre_k; c \leq \beta(t)$ 。

5) 增加从 $firing_i$ 状态开始的边, $\langle firing_i, dis_i \rangle$ 对应条件为 $p_m := p_m + post_m, \dots, p_n := p_n + post_n$ 。

6) 增加从 dis_i 状态开始的边, $\langle dis_i, enb_i \rangle$ 对应条件为 $p_i \geq pre_1, \dots, p_k \geq pre_k; c := 0; \langle dis_i, dis_i \rangle$, 对应条件为 $p_i < pre_i$ 。

7) 对每个迁移都进行上述转化, 最终形成整个 FTN 的转化。

图 4 是单个输入和输出库所迁移 t 转化为自动机图。

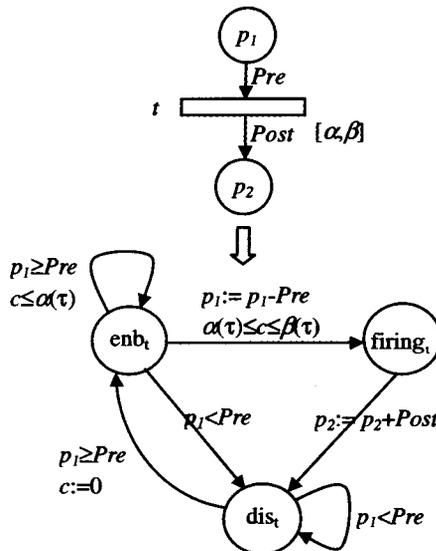


图 4 模糊 Petri 网转化为时间自动机

5.3 时间自动机的织入

织入时间自动机的形式化语义定义如下:

$$\begin{aligned}
 B &\xrightarrow{\alpha} B', res \xrightarrow{\mu, \beta, v} res' \wedge e \models \mu \wedge \beta \in \{a, *\}, \\
 (B, res, [T_i]_i) &\xrightarrow{\beta} (B', res', [T_i \leftarrow v'(T_i)]_i)
 \end{aligned}$$

B 代表系统行为模型; res 表示各个的时间方面模型; μ 表示对应的约束; β 表示对应的动作; v 表示对应的计时器;

$$v'(T_i) = \begin{cases} 0, & \text{当 } res \rightarrow res' \text{ 且 } T_i \in v \text{ 时} \\ T_i \clubsuit \epsilon, & \text{否则} \end{cases}$$

$\clubsuit \in \{+, \oplus\}$

6 举例

假设一个分布式多媒体系统, 其媒体数据存放在服务器, 服务器与客户端之间是不可靠的网络通道, 且数据是异步传输。当媒体数据客户端接收后, 便可进行播放。这个系统必须服从下列的时间约束条件:

T1: 服务器连续发送两帧数据之间的间隔 I 服从概率密度函数为 $X(I) = 0.02e^{-0.02I}$ 的指数分布, 其中 $I \geq 0$ 。

T2: 数据在传输中的延迟不超过以 30 为参数的指数分布, 否则视为丢失。通道允许数据丢失率为 0.1。

T3: 数据到达客户端之后, 进行播放需要延迟 5ms。

T4: 播放时数据等待 2ms 到 3ms 再进行处理, 播放器每秒处理 25 帧媒体数据进行显示, 并允许每帧有 5ms 的抖动。即处理每帧数据需要的时间为 (35, 40, 40, 45)。

上面的时间约束条件 T1、T2 属于不确定的时间方面, T3 属于确定的时间方面, 它们都可以利用 SQTl 来描述; T4 属于模糊时间方面, 可以用 FTN 表示。不确定方面有 T1 和 T2, 它们的表示及转换结果如下。

$$T1: [src_i \rightarrow (\diamond <_{\epsilon, 0.02} \rightarrow src_{i+1} \wedge \diamond =_{\epsilon, 0.02} src_{i+1})]$$

$$T2: [send_i \rightarrow (Oloss_i \vee \diamond <_{\epsilon, 0.1} \exp_{.30} rec_i)]$$

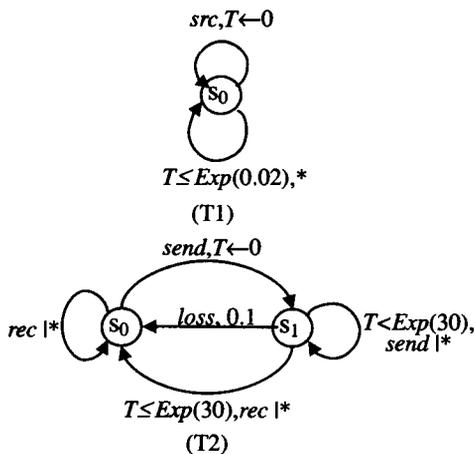


图5 T1和T2的时间自动机转化图

确定方面有 T3, 其表示及转换结果如下。

$$T3: (rec \rightarrow \diamond =_5 play)$$

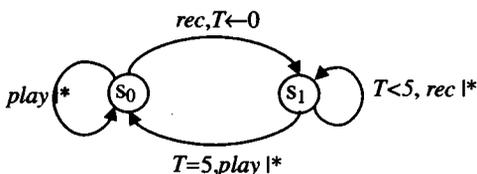


图6 T3的时间自动机转化图

模糊时间方面 T4 的表示及转换结果如下。

T4:

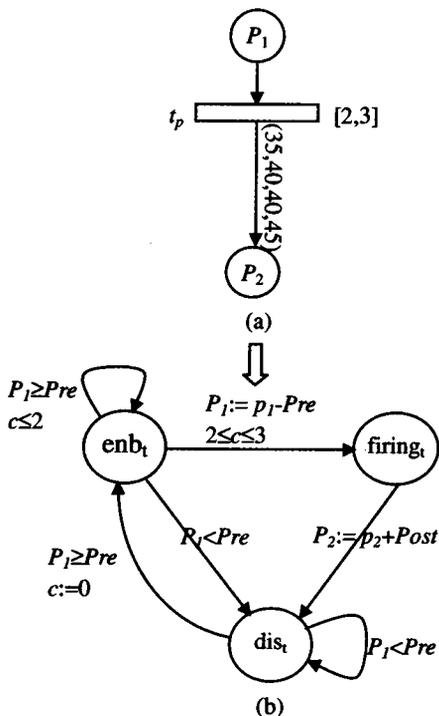


图7 T4的时间自动机转化图

织入并简化后的图如图8所示。

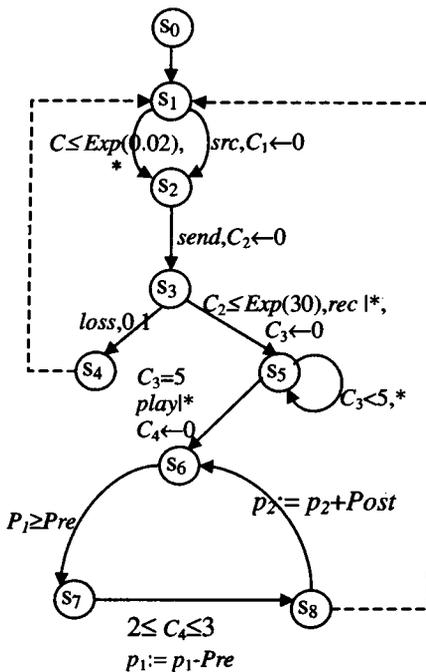


图8 实例的简化织入图

由于本文的方法涉及到状态空间爆炸问题,因此在转换为状态机后,最好进行状态的简化。另外,在模糊 Petri 网也可以进行 Petri 网的精简,可参考文[6]的精简方法。

结论 一般建模分布式实时系统是把时间关注分散在系统各个地方,从而使系统难于开发和维护。因此,我们引入了面向方面的思想设计分布式实时系统,来解决上述的问题。同时,本文也不同于其它的面向方面设计方法,我们把时间进一步分解为更细的时间方面,即确定时间方面、不确定时间方面和模糊时间方面;并且不同的方面根据不同的特点利用不同的设计语言实现,最后把各个时间方面织入系统,从而达到各个时间管理系统的实时特性。这里我们利用随机实时时序逻辑 SCTL 和模糊时间 Petri 网来实现分别表达确定时间方面与不确定时间方面和模糊时间方面,并通过转化为时间自动机来统一织入。

由于本文的方法涉及到状态空间爆炸问题,并限于篇幅,所以本文只举个简单的例子来说明。我们以后的工作就是考虑如何减少状态膨胀问题,并进一步完善这种设计方法。

参考文献

- 1 Kiczales G, Lamping J, Mendhekar A, et al. Aspect Oriented Programming. In: Aksit M, Matsuoka S. eds. Proceedings of the ECOOP, ser. Lecture Notes in Computer Science. Vol 1241. Berlin; Springer-Verlag, 1997. 220~242
- 2 Blair L, Blair G S, Andersen A. Separating Functional Behavior and Performance Constraints; Aspect-Oriented Specification[DB/OL]. Distributed Multimedia Research Group Report MPG-98-07, 1998
- 3 Lakas A, Blair G S, Chetwynd A. Specification and Verification of Real-Time Properties Using LOTOS and SCTL, In: Proceedings of the 8th International Workshop on Software Specification and Design, Paderborn, Germany, March 1996. 75~84
- 4 Murata M. Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets. Invited paper at the 17th International Conference on Application and Theory of Petri Nets, Osaka, Japan, LNCS Vol 1091. 11~28
- 5 Zhou Yi, Murata T. Modeling and Analysis of Distributed Multimedia Synchronization by Extended Fuzzy-timing Petri Nets. Transactions of the SDPS, 2001, 5(4): 23~37
- 6 Zhou Yi. Modeling and Analysis of Networked Real-time Systems via Fuzzy-timing Petri Nets; [Thesis for the degree of Ph. D]. Chicago; Electrical Engineering and Computer Science in the Graduate College of the University of Illinois, 2000
- 7 Cassez F, Roux O-H. From Time Petri Nets to Timed Automata, [rap. port]. R12003-4, Nantes, France; juin IRCCyN, 2003