

模糊时间 Petri 网及扩展的模糊时间 Petri 网 *

董玉梅 张立臣

(广东工业大学计算机学院 广州 510090)

摘要 网络实时系统对时间和 QoS 有严格要求,因此对设计、实现和测试的技术和工具都有很高的要求。模糊时间 Petri 网是将模糊集合理论用于不确定的或主观的时间信息的表示,在每个变迁上添加时间间隔约束对模糊时间 Petri 网进行扩展得到扩展的模糊时间 Petri 网,它是一种对网络实时系统进行建模和分析的形式化模型, Petri 网数学理论基础能保证并发系统的可靠性和正确性。本文详细地介绍了模糊时间 Petri 网及可扩展的时间 Petri 网,可扩展的时间 Petri 网的网精简方法,使用 design/CPN 仿真,及使用 EFTN 对多媒体同步建模,并探讨了进一步研究的方向。

关键词 形式化建模,网精简,实时系统,模糊时间 Petri 网,扩展的模糊时间 Petri 网

Fuzzy-Timing Petri Nets and Extended Fuzzy-Timing Petri Nets

DONG Yu-Mei ZHANG Li-Chen

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

Abstract Network-based real-time system has strict requirements for time and QoS, therefore techniques and tools helping engineers design, implement and test these applications are highly desirable. Fuzzy-Timing Petri Net (FTN) employ fuzzy set theory to express uncertain or subjective timing information, by each transition being associated with a firing interval, we get Extended Fuzzy-Timing Petri Net (EFTN). Extended Fuzzy-Timing Petri Net (EFTN) is a formal model for modeling and analysis of network-based real-time systems, Petri Net's mathematical theory basis will ensure the reliability and correctness of concurrent systems. This paper detailedly introduces the definition of FTN and EFTN, model check method, net-reduction technology, using design/CPN to simulate, modeling distributed multimedia synchronization by EFTN, and gives some directions of future work.

Keywords Formal modeling, Net-reducing, Real-time system, Fuzzy-timing Petri Net (FTN), Fuzzy-timing Petri Net (FTN)

1 模糊时间 Petri 网 (Fuzzy-Timing Petri Nets)

1.1 相关工作

Petri 网、时序逻辑和自动机都是并发系统建模的著名的形式化方法。在所有这些形式化方法中, Petri 网有严格的数学分析能力并且能保证并发系统的可靠性和正确性。现在已经有许多 Petri 网在时间方面的扩展,比如随机 Petri 网 (Stochastic Petri nets)^[1], Merlin 时间 Petri 网 (Merlin's Petri Nets)^[2], 对象组成 Petri 网 (Object Composition Petri Net)^[3], 时间流 Petri 网 (Time Stream Petri net)^[4]。随机 Petri 网中每个变迁与一个指数分布的随机变量相关联,表示从点燃可实施变迁点燃的延迟。如果几个变迁同时可实施,则具有最短延迟的变迁最先点燃。Merlin 时间 Petri 网中每个变迁 T 与时间间隔 $[t_{\min}, t_{\max}]$ 有关。假设在时间 t_0 可实施,则变迁 T 的点燃时间在 $t_0 + t_{\min}$ 和 $t_0 + t_{\max}$ 之间。

1.2 模糊时间 Petri 网的定义

为了处理实时系统中的时间不确定性问题^[5], Murata 提出了模糊时间高级 Petri 网 (Fuzzy-Timing High-Level Petri Nets)^[6], 将模糊集合理论用于不确定的或主观的时间信息的表示。FTHN 的主要特性是如下四个模糊集合理论的时间函数: 模糊时间戳 (fuzzy timestamp), 模糊可实施时间 (fuzzy enabling time), 模糊发生时间 (fuzzy occurrence time) 和模糊延迟 (fuzzy delay)。模糊时间戳 $\pi(\tau)$ 是个模糊时间函数或者

概率分布,表示一个特定的令牌在时间 τ 到达特定位置的概率。如果 $\pi(\tau) = 0.5$,说明它在时间 τ 到达的概率为 0.5。从变迁 t 到位置 p 的每一个弧 (t, p) 和模糊延迟 $d_{tp}(\tau)$ 相关。为了简单,用四元组 $(\pi_1, \pi_2, \pi_3, \pi_4)$ 定义的梯形或三角形的概率分布来表示模式时间函数,其中高度 h_i 的默认值是 1, 三角形是梯形中 $\pi_2 = \pi_3$ 的特殊形式。

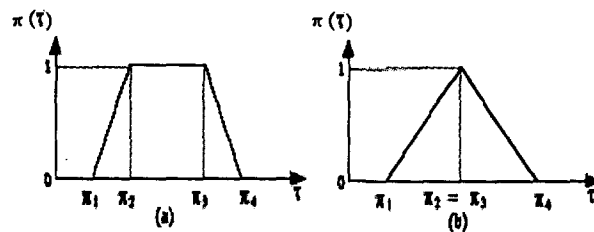


图 1 梯形和三角形概率分布

模糊时间高级 Petri 网的静态结构是一个九元组, $N = (\Sigma, P, T, A, C, D, CT, FT, F)$ 其中

- (I) Σ 是有限的类型集合,称为颜色集合
- (II) P 是有限的库所的集合
- (III) T 是有限的变迁的集合
- (IV) $A \subseteq (P \times T) \cup (T \times P)$ 是弧的集合
- (V) $C \in P \rightarrow \Sigma$ 是个颜色函数
- (VI) D 是所有模糊延迟 $d_{tp}(\tau)$ 的集合,每个延迟与 arcs

* 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基金项目 (No. Z03024) 基金资助。董玉梅 硕士研究生,主研实时系统。张立臣 博士,教授,主研实时系统。

$\subseteq(T \times P)$ 相关

(VII) $CT = \{ \langle p, v \rangle \mid p \in Pand v \subseteq P \}$ 是所有的颜色令牌的集合

(VIII) FT 是所有模糊时间戳的集合。时间戳是一个从时间范围 TS 到实数间隔 $[0, 1]$ 的映射函数

(IX) F 是变迁函数, 表示当事件的变迁点燃时令牌的移动情况。

2 扩展的模糊时间 Petri 网 (Extended Fuzzy-timing Petri Nets)

在每个变迁上添加时间间隔约束 $p[a, \beta]$ 对模糊时间 Petri 网进行扩展得到扩展的模糊时间 Petri 网^[7]。

2.1 扩展的模糊时间 Petri 网定义

通过把 FTN 和 Merlin 时间 Petri 网相结合得到 FTN 的扩展。当扩展的模糊时间 Petri 网模型的 $d_p(\tau)$ 默认值是 $(0, 0, 0, 0)$ 的时候就变成了 FTN 模型, 另外还有一个附加的函数 CT:

$$T \rightarrow Q^+ \times Q^+ \times (Q^+ \cup \infty),$$

即从变迁集合 T 到点燃瞬间的映射是个概率分布, 就是说每个变迁与点燃瞬间 $p[a, \beta]$ 相对应, 默认值是 $1[0, 0]$ (一个变迁使能之后马上点燃)。如果一个变迁 t 在时刻 τ 使能, 那么 t 在 $\tau + \alpha$ 之前不能点燃, t 必须在 $\tau + \beta$ 时刻或之前点燃。概率 $p \in [0, 1]$ 。如果变迁 T 与其他变迁没有冲突, 那么 $p = 1$ 。当想为冲突的变迁赋予不同机率时, $p < 1$ 。比如说变迁 t_1 和 t_2 有冲突, t_1 以 99% 的机率点燃, t_2 以 1% 的机率点燃, 那么让 $p_1 = 0.99, p_2 = 0.01$ 。变迁点燃本身是一个原子过程, 不占用时间。为变迁赋予点燃瞬间概率的目的在于当发生冲突时给与点燃的可能性和优先级。这在实时系统建模中很有用。

2.2 扩展的模糊时间 Petri 网的模糊事件戳更新

1) 计算每个变迁 t 模糊使能时间。假设变迁 t 需要 n 个令牌才能点燃, 它们的模糊时间戳是 $\pi_i(\tau), i = 1, 2, \dots, n$, 那么变迁 t 模糊使能时间 $e_t(\tau) = \text{latest} \{ \pi_i(\tau), i = 1, 2, \dots, n \}$, latest 运算符表示“最晚到达/最低概率的分布”, 每个变迁需等待最晚到达的令牌。

2) 计算下一个变迁发生的概率分布。变迁 t 的模糊发生事件 $o_t(\tau)$ 是变迁 t 开始点燃时间的概率分布。当有 m 个变迁发生冲突时, 他们的模糊使能时间 $e_i(\tau), i = 1, 2, \dots, t, \dots, m$ 变迁 t 的模糊使能时间为 $e_t(\tau)$, 变迁 T 的模糊发生时间为

$$o_t(\tau) = \{ e_t(\tau) \oplus p_1(\alpha, \alpha, \beta, \beta),$$

$$\text{earliest} \{ e_i(\tau) \oplus p_i(\alpha, \alpha, \beta, \beta), i = 1, 2, \dots, t, \dots, m \},$$

earliest 运算符是从 m 个分布中求“最早到达/最大概率分布”。当变迁 T 与其他变迁没有冲突的时候, 其中 $CT(t_i) = p_i[\alpha, \beta]$, 那么模糊发生时间为 $o_t(\tau) = e_t(\tau) \oplus (\alpha, \alpha, \beta, \beta)$ 。

3 扩展的模糊时间 Petri 网的性能分析技术-网精简

当软件系统的大小成线性增长时, 分析这个系统所用的时间和空间可能成指数增长。实时系统的验证比非实时系统更复杂, 因为实时系统的正确性不仅依赖于它们的功能而且取决于它们的时间限制。在所有说明和验证实时系统的形式化语言中都存在状态爆炸问题。和其他形式化语言相比, 时间扩展的 Petri 网比非时间扩展的 Petri 网分析的复杂度要高。在大规模实时分布式系统中使用性能分析技术是解决“状态爆炸”问题的切实可行的途径。网精简技术是 Petri 网中解决状态爆炸问题的一个重要技术, 它致力于减小 Petri 网模型的规模, 同时保证并发的特性比如活性, 安全性和有界

性。

精简规则 1(变迁的向后融合) 如果存在一个位置 p 满足如下条件变迁集合 TA 就能向后融合为变迁集合 TB 。1) $\bullet p = TA$ 和 $p \bullet = TB$ (TA/TB 是位置 p 的输入变迁和输出变迁) 2) 位置 p 开始没有令牌 3) 对于每个变迁 $tb \in TB, \bullet tb = \{ p \}$ 并且 $p \notin tb \bullet$ (p 是 tb 的唯一的输入位置但不是输出位置) 4) $TA \cap TB \bullet = \phi$ (对于每个 $ta \in TA$ 和每个 $tb \in TB, ta$ 和 tb 没有共同的输出位置) 5) 对于每个变迁 $tb \in TB,$

$$[\min(eb1, \dots, ebn), \min(lb1, \dots, lbn)] \cap [eb, lb] \neq \phi$$

(每个变迁 tb 都有机会点燃) 6) 对于每个变迁 $tb \in TB$, 如果 $|tb \bullet| > 1$, 那么 $tb = lb$ 。

证明: 假设变迁 ta 的模糊发生时间为 $o_{ta}(\tau)$, 那么 (1) 对于 EFTN N , 令牌到达位置 p 的模糊时间分配为 $\pi_p(\tau) = o_{ta}(\tau) \oplus da(\tau)$ 。因为 p 是 TB 中所有变迁的唯一输入位置, p 除了 TB 中的变迁之外没有其他的输出变迁, 每个变迁 tbi 在时间 $e_{tbi}(\tau) = \pi_p(\tau)$ 使能, 因为 TB 中的所有变迁都冲突, 变迁 tbi 的模糊发生时间为 $o_{tbi}(\tau) = (e_{tbi}(\tau) \oplus [eb1, lb1]) \cap \text{earliest} \{ (e_{tbi}(\tau) \oplus [eb1, lb1]), \dots, (e_{tbn}(\tau) \oplus [ebn, lbn]) \} = o_{ta}(\tau) \oplus da(\tau) \oplus [eb1, lb1] \cap [\min(eb1, \dots, ebn), \min(lb1, \dots, lbn)]$ 所以, 令牌到达位置 $pb1$ 的模糊时间分布为 $\pi_{pb1}(\tau) = o_{tbi}(\tau) \oplus db1(\tau) = o_{ta}(\tau) \oplus da(\tau) \oplus db1(\tau) \oplus [eb1, lb1] \cap [\min(eb1, lb1), \dots, \min(ebn, lbn)]$

(2) 对于 EFTN N_R , 集合 TF 中的每个变迁与 EFTN N 中的变迁 ta 有相同的模糊发生时间, 所以 $o_{t_i1}(\tau) = o_{ta}(\tau)$, 令牌到达位置 $pb1$ 的模糊时间分布为 $\pi'_{pb1}(\tau) = o_{t_i1}(\tau) \oplus df1(\tau) = o_{ta}(\tau) \oplus da(\tau) \oplus [eb1, lb1] \cap [\min(eb1, \dots, ebn), \min(lb1, \dots, lbn)] = \pi_{pb1}(\tau)$ 在 EFTN N 相似的, EFTN N_R 和 EFTN N 中令牌到达位置 pbs 的模糊时间分布相等。所以 EFTN N 和 N_R 在时间上相同。因为位置 pa 开始时没有令牌, 所以 EFTN N 和 N_R 的死锁等同。

精简规则 2(变迁的并行融合) 如果存在位置 pa 和 pb 满足下面条件那么两个变迁 t_1 和 t_2 也是可融合的。1) $\bullet t_1 = \bullet t_2 = \{ pa \}$, 并且 $t_1 \bullet = t_2 \bullet = \{ pb \}$ 2) $[eb1, lb1] \cap Itf \neq \phi$ 并且 $[eb2, lb2] \cap Itf \neq \phi$, 其中 $Itf = [\min(eb1, eb2), \min(lb1, lb2)]$ (t_1 和 t_2 都有机会点燃) 3) 时间间隔 $([eb1, lb1] \cap Itf) \oplus db1(\tau)$ 和 $([eb2, lb2] \cap Itf) \oplus db2(\tau)$ 是重叠的, $([eb1, lb1] \cap Itf) \oplus db1(\tau)$ 是相对于令牌到达位置 pa 的时间, 点燃变迁 t_1 令牌到达位置 pb 的最早时间, $([eb2, lb2] \cap Itf) \oplus db2(\tau)$ 是相对于令牌到达位置 pa 的时间, 点燃变迁 t_2 令牌到达位置 pb 的最早时间。(点燃 t_1 和 t_2 令牌到达 pb 的时间间隔) 4) 如果 $pa \bullet \neq \{ t_1, t_2 \}$ (pa 的输出变迁不是 t_1 和 t_2), 就假设变迁 t 是在 t_1 和 t_2 之间具有最早点燃时间的变迁 ($e_{bt} = \min[eb1, eb2]$), 令牌到达 pb 的最早时间和最晚时间都需要通过 t 的点燃来获取。

证明: 假设令牌到达位置 pa 的模糊时间分布为 $\pi_{pa}(\tau)$, 所以 (1) 对于 EFTN N , 因为 pa 是变迁 t_1 和 t_2 唯一的输入变迁, 变迁 t_1 和 t_2 在 $e_{t1}(\tau) = e_{t2}(\tau) = \pi_{pa}(\tau)$ 使能, 因为变迁 t_1 和 t_2 发生冲突, 所以变迁 t_1 的模糊发生时间为 $o_{t1}(\tau) = (e_{t1}(\tau) \oplus [eb1, lb1]) \cap \text{earliest} \{ (e_{t1}(\tau) \oplus [eb1, lb1]), (e_{t2}(\tau) \oplus [eb1, lb2]) \} = \pi_{pa}(\tau) \oplus [eb1, lb1] \cap [\min(eb1, eb2), \min(lb1, lb2)]$ 所以, 因为变迁 t_1 的点燃令牌到达位置 pb 的模糊时间分布为 $\pi_{pb1}(\tau) = o_{t1}(\tau) \oplus db1(\tau) = \pi_{pa}(\tau) \oplus db1(\tau) \oplus ([eb1, lb1] \cap [\min(eb1, eb2), \min(lb1, lb2)])$ 。相似的变迁 t_2 的模糊发生时间为 $o_{t2}(\tau) = \pi_{pa}(\tau) \oplus ([eb2, lb2]) \cap [\min(eb1, eb2), \min(lb1, lb2)]$, 由于变迁 t_2 的点燃令牌到达位置 pb 的模糊时间分布为 $\pi_{pb2}(\tau) = o_{t2}(\tau) \oplus db2(\tau) \oplus ([eb2,$

的模糊时间分布为 $\pi_p(\tau) = o_i(\tau) \oplus dai(\tau)$ 。所以变迁 tb 的使能时间为 $e_b(\tau) = latest\{\pi_{p1}(\tau), \dots, \pi_{pn}(\tau), \pi_{pbj}(\tau)\} = latest\{o_i(\tau) \oplus latest\{dai(\tau), \dots, dan(\tau)\}, \pi_{pbj}(\tau)\}$ ，如果变迁 tb 有输入位置 pbj ，则 $\pi_{pbj}(\tau)$ 是令牌到达位置 pbj 的模糊时间分布。

(2) 对于 EFTN N_R ，由于变迁 t 的点燃令牌到达位置 pf 的模糊时间分布为 $\pi_{pf}(\tau) = o_i(\tau) \oplus latest\{dai(\tau), \dots, dan(\tau)\}$ 。变迁 tb 的使能时间为 $e_b(\tau) = latest\{\pi_{pf}(\tau), \pi_{pbj}(\tau)\} = latest\{o_i(\tau) \oplus latest\{dai(\tau), \dots, dan(\tau)\}, \pi_{pbj}(\tau)\}$ ，这和 EFTN N 中的 $e_b(\tau)$ 相等。

如果每个位置初始是有一个令牌，由于变迁 t 的点燃产生的令牌使变迁 tb 使能，证明的方式和前面位置 pi 中没有令牌的情形一样；在 EFTN 中，对于位置 pi 中初始时含有的令牌，用来使能变迁 tb ，变迁 tb 的使能时间为 $e'_{tb}(\tau) = latest\{(0,0,0,0), \dots, (0,0,0,0), \pi_{pbj}(\tau)\} = \pi_{pbj}(\tau)$ 。对应 EFTN N 中的这些令牌，EFTN N_R 中位置 pf 开始时含有一个令牌，用这个令牌使变迁 t 使能的时间为 $latest\{(0,0,0,0), \dots, (0,0,0,0), \pi_{pbj}(\tau)\} = \pi_{pbj}(\tau)$ ，这和 EFTN N 中的 $e'_{tb}(\tau)$ 相等。得出结论 EFTN N 和 EFTN N_R 在时间和死锁方面相等。

精简规则 7(冗余位置的删除) 如果位置 p 开始时没有令牌而且没有输入变迁，那么 p 和所有的输出变迁都可以被移除。 N_R 是由 EFTN N 冗余位置的删除规则得到的，那么 N_R 和 N 在时间和死锁方面等价。

证明：在 EFTN N 中，位置 p 开始时没有令牌而且没有输入变迁，所以位置 p 永远不可能含有一个令牌而且位置 p 的所有输出变迁都不可能使能和点燃。在 EFTN N_R 中移除它们，时间和死锁特性并不会改变。

4 扩展模糊时间 Petri 网的性能分析技术—使用 Design/CPN 进行仿真

基于形式化建模的仿真可以用来预测软件设计的行为。由于 Design/CPN 具有普遍性，图形界面和直观性，以及支持时间分布的代码，使用 Design/CPN 进行仿真。

4.1 仿真模糊时间功能

使用类似于模糊变量的单值仿真对 EFTN 模型的模糊延迟进行仿真。定义一个函数来产生 Design/CPN 中四边形概率分布标准值：给定一个模糊延迟 $D(\tau) = (a, b, c, d)$ ，定义一个函数 FUZZY(a, b, c, d) 来产生四边形概率分布的单个延迟值 (a, b, c, d) 来代表模糊延迟。Design/CPN 中 FUZZY(a, b, c, d) 重复以下的过程直到得到一个延迟值：产生一个时间间隔 $[a, d]$ 中的随机时间值 $atime$ ；如果 $atime$ 在时间间隔 $[b, c]$ ， $atime$ 就将作为延迟值，因为时间间隔 $[b, c]$ 内的概率是 1；如果 $atime$ 在时间间隔 $[a, b]$ 或 $[c, d]$ ，产生一个 $(0, 1)$ 的随机值 V 并且计算四边形分布 (a, b, c, d) 中的概率 $D(atime)$ 。只有当 $D(atime) \geq V$ 时， $atime$ 才作为延迟值。

4.2 将扩展的模糊时间 Petri 网进行仿真

在 EFTN 模型中，一个变迁可能与点燃间隔和点燃概率相关联。Design/CPN 默认不支持这些特性。在 Design/CPN 中，时间延迟可能和变迁的输出弧和变迁自己相关联。但是，Design/CPN 中与变迁相关联的延迟实际上时变迁点燃的延迟。EFTN 中变迁点燃间隔是指从变迁使能开始到变迁点燃的时间间隔。EFTN 模型中变迁点燃的时间为 0。为了使用 Design/CPN 进行仿真，通过添加一些多余的变迁，位置和代码段把 EFTN 模型转化为 Design/CPN 模型。图 2a 阐明了有冲突的变迁的转化。在转化模型中，产生一个 $[0, 1]$ 内的随机数，如果这个随机数小于 0.8 则变迁 $t1$ 点燃，否则就是变迁 $t2$ 点燃。这个转化的模型模拟了 EFTN 模型中有点燃概率的变迁 $t1$ 或 $t2$ 。图 2b 阐明了变迁点燃间隔的转化。在转化的模型中，位置 $P1$ 和 $P2$ 中的每个令牌都有一个唯一的标识号。当位置 $P1, P2$ 分别被 $token_i, token_j$ 使能，在时间间隔 $[d1, d2]$ 一个令牌将放到位置 $t1_ready$ 。当令牌到达位置 $t1_ready$ 时，如果 $token_i$ 和 $token_j$ 仍然在位置 $P1$ 和 $P2$ 中时，变迁 $t1$ 将被点燃。EFTN 模型中变迁 $t2$ 的点燃过程和 $t1$ 的点燃过程相类似。

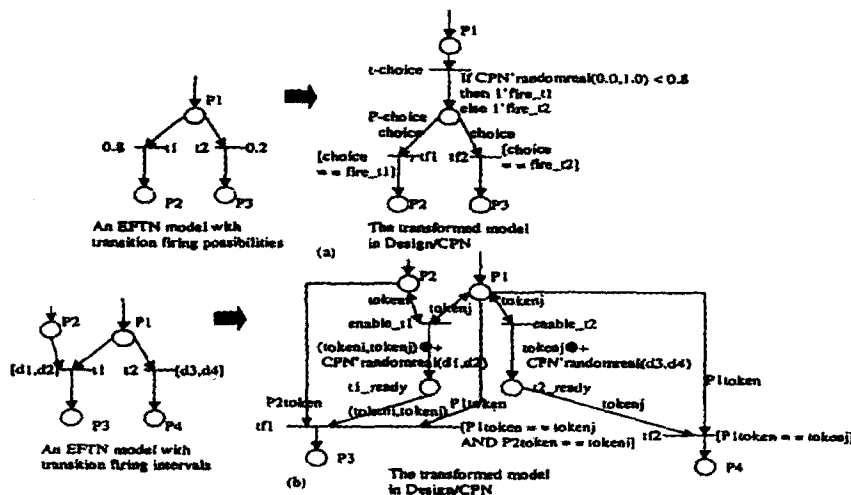


图 2 将 EFTN 模型中的变迁点燃概率和时间间隔转化成 Design/CPN

5 将扩展的模糊时间 Petri 网模型和分析应用于分布式多媒体同步^[7]

分布式多媒体系统支持数据库中或网络中的声音、视频、文本和数字数据的整合与协同。多媒体同步指数据流的时间组成。数据流可定义为信息单元的时间序列，比如说视频流就是视频帧的时间序列。根据同步粒度，多媒体时间播放模

型又被划分为粗颗粒模型和细颗粒模型。粗颗粒模型仅仅说明对象之间的同步，同步仅发生在对象的开始和结束，所以粗颗粒模型中的同步是对象级别上的同步。细颗粒同步的单位是信息单元，信息单元语义上与严格的时间需求相关，比如延迟和抖动。所以细颗粒模型可分为流内同步和流间同步。流内同步说明了每个多媒体流内信息单元的时间限制。连续播放的媒体之间的严格同步(比如声音流和视频流之间的同步)

属于流间同步。为了保证时间组成上的 QoS 需求,信息单元级别上的现场流的细粒度同步应由时间模型提供。

分布式多媒体同步包含异构的源和媒体之间的异步通信。不是所有的媒体分段都有事先知道的播放时间间隔,几个分布的、独立的源之间的多媒体同步可能包含不确定的时间需求,所以需要—个时间模型来处理这些不确定性。

Petri 网这种建模技术适合于描述并发、异步和分布式系统,它不仅提供了图形表示而且提供了分析。OCPN(Object Composition Petri Net 对象组成 Petri 网)和 TSPN(Time Stream Petri Net 时间流 Petri 网)是用来描述多媒体同步的时间需求的经典模型。OCPN 是对象级别的模型,位置与资源以及持续时间相关。持续时间是对象固定的播放时间,由对象的位置决定。变迁表示对象之间的同步点。OCPN 可以说明多媒体对象之间最基本的关系(之前,相遇,重叠,持续,开始,完成,相等,间隔)。在 OCPN 中,假设多媒体对象的播放时间事先知道并且固定。TSPN 是为数据同步建模的细粒度时间模型。在 TSPN 中,离开位置的弧用一个三元组 $[a, \eta, \beta]$ 表示, a, η, β 分别表示最早点燃时间,名义上的点燃时间以及最晚点燃时间。这可以用来定义最大允许的抖动(每个信息单元的时间偏移)。每个位置都有一个相对应的多媒体信息单元处理时间点。如果一幅图像播放的速度是每秒中 10 幅图,一个视频对象允许的最大抖动为 10ms,那么标识在离开位置的弧就是(90,100,110)。在 TSPN 中,每个变迁都被映射成一种点燃规则。三种基本的变迁点燃规则是强—或(strong-or),弱—与(weak-and),控制(master)。强—或规则是指同步是由最早到达的流驱动的,晚到的流的播放将被禁

止。弱—与规则是指同步是由最晚到达的流驱动的。控制规则是指同步是由控制流驱动的。到目前为止,TSPN 是多媒体同步最有力的工具。

5.1 使用 EFTN 为分布式多媒体同步建模和分析

EFTN 模型可以达到和 TSPN 相同的性能,它们都是细粒度的时间模型,都支持模糊的概念并且证明了满足 QOS 的可能性。从一系列相对时间关系出发,描述一步步建立细粒度 EFTN 模型的过程。

第一步,建立对象级别的 EFTN 模型的块

1)将 Master 变迁点燃规则模型化:TSPN 中的 Master 策略可以被 EFTN 模拟。如图 3 所示,定义 Master 流可能等待的最大时间,只有当 Master 流的等待时间超过限制,才能执行同步,并且其他的流被截止。

2)将 strong-or 变迁点燃规则模型化:TSPN 中的 strong-or 策略可以被 EFTN 模拟。EFTN 模型甚至更强大,因为它能说明最早到达的流可以等待的最大时间。只有当最早到达的流的等待时间超过限制,才能执行同步,并且其他的流被截止。

3)基本时间间隔关系的 EFTN 模型:EFTN 模型也可以模拟基本时间间隔关系为之前,相遇,重叠,持续,开始,完成,相等,间隔。如果 X 和 Y 之间没有选择 master 对象,而且限制值等于最早到达的流可以等待的时间长度,当最早到达的流完成或者等待超出时间限制,那么晚到的流将被截止。如果选中 X 作为 master 对象,如果 Y 在 X 完成之前或者等待超出时间限制之前没能完成,那么 Y 将被截止。如果想让流之间互相等待,可以把时间限制设置成 $+\infty$ 。

A TSPN model with master transition firing rule

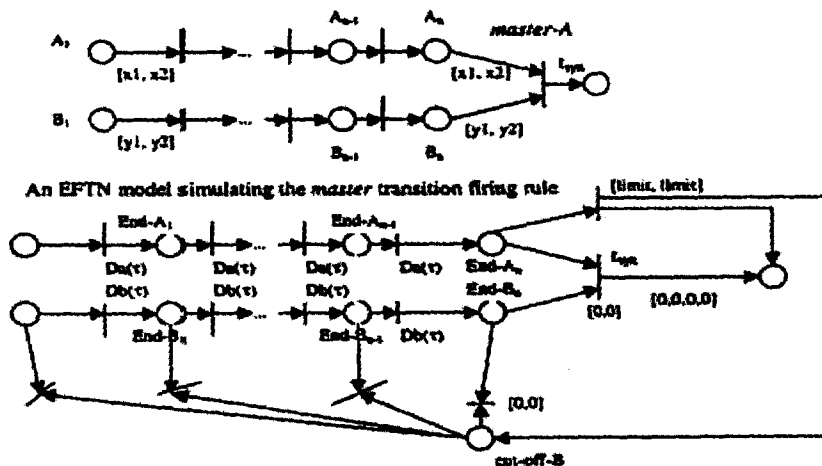


图 3 使用 EFTN 对 master 变迁点燃规则建模

第二步,将对象级别模型分解成细粒度时间模型

当得到一个对象级别模型之后,为了满足 QoS 的需求,需要把对象分解成信息单元来为流内同步和流间同步建模。流间同步主要关注每个信息单元最大允许的抖动。在 EFTN 中,一个信息处理变迁的点燃发生在处理相对应的多媒体信息单元的始点。假设一个信息单元的名义点燃时间为 n ,最大允许偏移量为 $\pm \eta$ 。离开变迁进入相应 End-Information-unit 的弧的持续时间为 $(n - \eta, n, n + \eta)$ 。

第三步,计算 QoS 需求被满足的可能性

多媒体同步中的重要问题就是检查 QoS 的需求有没有得到满足。举例来说,需要确定流间同步在哪里插入同步点。

这需要满足最大允许偏移量的需求。可以使用 FTML 公式 ϕ 定义 QoS 需求,并且计算 EFTN 模型是否满足 ϕ 。如果计算得出的满足 QoS 需求的概率小于最低限度,需要进一步将同步粒度精化。

结论和发展方向 扩展的模糊时间 Petri 网(EFTN)是一种对网络实时系统进行建模和分析的形式化模型, Petri 网数学理论基础能保证并发系统的可靠性和正确性。本文介绍了扩展的模糊时间 Petri 网的定义,模型检验方法,仿真工具,网精简技术和应用领域。EFTN 可以捕获时间的不确定性并且可以进行有效的性能分析,所以 EFTN 适合为实时系统建模。还要进一步做的工作有:

1. 对扩展的模糊时间 Petri 网进行进一步的扩展,包括把模糊逻辑和各种时序逻辑、Petri 网结合起来,例如对时序逻辑的可能性分析,可以应用实时计算树逻辑(Real-Time Computation Tree Logic),这项技术可以用于有限状态并发系统的状态转移图。

2. 把扩展的模糊时间 Petri 网和 UML 等系统分析与建模工具结合起来,使扩展的模糊时间 Petri 网可以应用更复杂的实时系统的建模。

3. 进一步应用扩展的模糊时间 Petri 网与复杂的实时系统的调度、网络 QoS 的分析、多媒体系统的建模, CIMS 系统及 P2P 系统等。

参 考 文 献

- 1 Marsan M A, Balbo G, Conte G. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM trans. on computer system*, May 1984, 2(2)
- 2 Merlin P, Faber D. Recoverability of communication protocols-implications of a theoretical study. *IEEE Trans. Communications*,

Sept. 1976

- 3 Little T D C, Ghafoor A. Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks. *IEEE Computer*. Oct. 1991, 24(1)
- 4 Diaz M, Senac P. Time Stream Petri Nets: A Model for Multimedia Streams Synchronization. In: *Proc. of the First Intl. Conf. on Multi-Media Modeling*, Nov. 1993, Singapore
- 5 Valette R, Cardoso J, Dubois D. Monitoring Manufacturing Systems by Means of Petri Nets with Imprecise Marking. *IEEE International Symposium on Intelligent Control*, Albany, N. Y., USA
- 6 Murata T. Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets. *Application and Theory of Petri Nets, Lecture Notes in Computer Science*, Vol. 1091, Springer-Verlag, New York, June 1996
- 7 Zhou Y, Murata T. Fuzzy-Timing Petri Net Model for Distributed Multimedia Synchronization. In: *Proc. of the 1998 IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC'98)*, La Jolla, Calif., Oct. 1998

(上接第 211 页)

语法和语义的正确。语法的正确性可以由系统的语法检查保证;语义上的正确性虽然从原理上讲,系统不能完成保证,但是可以提供一些手段帮助集成人员进行语义上的检查。

参 考 文 献

- 1 Carriero N, Gelernter D. Coordination languages and their significance. *Communications of the ACM*, Feb. 1992, 35(2): 97~107
- 2 Papadopoulos G A, Arbab F. Coordination models and languages. *Software Engineering (SEN) SEN-R9834*, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Netherlands, Dec. 1998
- 3 Carriero N, Gelernter D. Linda in Context, *Communications of the ACM*, April 1989, 32(4)
- 4 Arbab F, Herman I, Spilling P. An overview of Manifold and its implementation. *Concurrency: Practice and Experience*, Feb. 1993, 5(1): 23~70
- 5 Magee J, Dulay N, Eisenbach S, Kramer J. Specifying distributed software architectures. In: *Proc. of the Fifth European SoftwareEngineering Conf. Sep. 1995*
- 6 Shaw M, DeLine R, Klein D, et al. Abstractions for software architecture and tools to support them. *IEEE Transaction on Software Engineering*, April 1995, 21(4): 314~335
- 7 Medvidovic N, Taylor R N. A classification and comparison framework for software architecture description languages. *IEEE Transaction on Software Engineering*, Jan. 2000, 26(1): 70~93
- 8 Shaw M. Procedure Calls are the Assembly Language of System

Interconnection; Connectors Deserve First Class Status. In: *Proc. of the Workshop on Studies of Software Design*, May 1993

- 9 Shaw M. Architectural Issues in Software Reuse; It's Not Just the Functionality, It's the Packaging. In: *Proc IEEE Symposium on Software Reusability*, April 1995
- 10 Mehta N R, Medvidovic N, Phadke S. Towards a Taxonomy of Software Connectors. *ICSE '00*, Limerick, Ireland, May 2000
- 11 Kramer J, Magee J, Finkelstein A. A Constructive Approach to the Design of Distributed Systems, Tenth Intl. Conf. on Distributed Computing Systems (ICDCS'90), Paris, France, 26 May - 1 June, IEEE Press, 1990. 580~587
- 12 Barbacci M R, Weinstock C B, Doubleday D L, Gardner M J, Lichota R W, Durra; A Structure Description Language for Developing Distributed Applications, *Software Engineering Journal*, IEEE, March 1996. 83~94
- 13 Bergstra J A, Klint P. The TOOLBUS Coordination Architecture. In: *First Intl. Conf. on Coordination Models, Languages and Applications (Coordination'96)*, Cesena, Italy, April, LNCS, Springer Verlag, 1996. 75~88
- 14 Garlan D, Allen R, Ockerbloom J. Architectural mismatch, or why it's hard to build systems out of existing parts. In: *Proc. of the Seventeenth Intl. Conf. on Software Engineering*, Seattle, Washington, April 1995. 85~179
- 15 Gamma E, Helm R, Johnson R, Vlissides J. *Design Patterns - Elements of Reusable Object-Oriented Software*. Boston: Addison-Wesley. 1995