

基于 CPN 的信息系统资源配置动态模型的建立

颜志军, 甘仞初

(北京理工大学管理与经济学院, 北京 100081)

摘要: 针对信息资源配置问题提出了一种基于有色 Petri 网的规范化建模方法, 并对整个建模过程进行了详细的描述。提出了系统层次模型结构和执行图的概念, 对执行图和 Petri 网之间的转换过程进行了分析、定义, 从而减少了大系统建模的复杂程度和工作量。利用相关的计算机工具, 可以方便地实现系统建模和对信息资源配置情况的评价、分析和调整。

关键词: 信息系统; CPN; 资源配置; 性能评价

中图分类号: C939; O 233

文献标识码: A

文章编号: 1007-9807(2002)01-0076-09

0 引言

信息系统资源配置问题是信息系统开发过程中面临的一个主要问题, 合理的资源配置方式能保证信息系统最大限度地发挥作用, 以最快、最可靠的方式满足用户的信息需求。自 60 年代末开始发展的传统信息系统开发方法面对目前复杂的信息系统环境, 已难以实现资源的合理配置。因此, 提出一种能有效分析、评价系统资源配置状况的方法已迫在眉睫^[1]。

目前, 有不少学者在信息系统资源模型方面开展了许多工作, 如文[3]提出的信息系统资源配置静态模型; Daneva 提出的 Benchmarking 方法^[4]; Opdahl 提出的利用功能模块层次图和排队模型来估计系统的性能状况^[5]; Gunnar B rataas 提出的利用 Structure and Performance (SP) 模型来模拟资源流动进而分析系统性能^[6]; Frescha 提出的 PRM 方法^[7], 等等。这些方法在一定程度上实现了信息系统资源建模, 但它们都在不同程度上存在无法描述系统资源竞争和同步现象、无法描述网络系统等缺点。

文中提出的基于 CPN 的信息系统资源配置模型能够实现信息系统资源建模, 对信息系统的行为特性进行描述, 表征系统在运行状态下的资

源状况, 避免上述建模方法中存在的缺陷。模型的建立过程是基于有色 Petri 网来进行, 有色 Petri 网 (CPN) 是 Kurt Jensen 在基本 Petri 网基础上提出的一种扩展 Petri 网模型^[8]。它是一种对复杂系统进行检验和仿真的有色建模语言, 通过将相似部件或行为进行折叠, CPN 可提供一种以紧凑、简明的方式来表征一个具有若干相似部件的大系统的能力, 从而解决系统模型规模不可控制的问题。因此, CPN 除具有普通 PN 所具有的可描述系统动态特性、表征系统竞争和冲突现象等优点外, 还具有模型结构简单、可重用性强等优势。

文中提出了层次体系结构的系统资源配置动态模型 (见图 1), 并对建模过程进行了规范化从而使其能运用于复杂大系统的建模。采用层次结构模型能够使对信息资源调用情况的描述变得一目了然, 并直观反映信息系统对信息资源操作自顶向下的传递关系, 体现各功能模块对信息资源的占用情况。同时, 层次模型还能给用户较好的可扩展性, 用户完全可以根据系统建模的目标和应用系统的实际情况对层次模型进行不同级别的扩展, 保证模型能最大限度地与实际系统相吻合。层次模型良好的扩展性也保证了它具有更大的灵活性, 可以按用户建模的不同目的来进行组

合、改变 信息系统层次模型通过对应用系统、数据管理和资源操作等不同层次的分析建模而组合成一个整体模型。应用处理层是系统模型的顶层, 可以通过执行图实现自动生成; 数据管理层主要模拟系统功能对数据的不同操作要求, 并通过对数据管理需求的分类来得到不同的页; 资源操作层模拟功能操作对实际资源的占用情况, 在这个层次上, 用户可以根据需要建立不同细节的模型, 对原体系结构进行不同程度的扩展

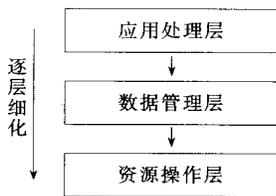


图 1 模型层次体系结构

文中采用的建模工具 Design/CPN 是用于创建、编辑 CPN 模型的计算机图形工具^[9]。它提供了多种工具来实现对 CPN 模型的分析, 并通过 SML^[10] 语言来实现对模型仿真的细节控制。Design/CPN 的最大好处在于: 当网模型结合了有色托肯、延时变迁和层次性等特点后将提供一种坚实基础来支撑对大型分布式系统的建模分析, 而这种模型也很容易为了不同的目的而进行组合和改变^[11]。

1 动态模型的建立

本节将提出执行图的概念, 并以一个租车系统为例来详述整个模型的建立过程

1.1 问题描述

例 1 用户租车系统: 租车系统是为用户提供租车服务。用户可以通过提前预约的方式或直接填写租借单来进行租车活动。当采用预约方式时, 用户首先填写预约单, 然后在预约日期到租车点租车并填写租车单。同时, 用户可以在租车前取消预约, 若用户直接到租车点租车, 则首先填写客户登记表, 并在填写租车单后可以租车。用户凭租车单到车库提车, 在使用一段时间后, 用户到租车点还车并结帐离开。

假设 租车公司在北京、广州、上海均设有分公司, 并且存在异地预约的情况。即北京用户可能

在北京分公司预约某天广州或上海的车辆, 所以在每地的用户都可查询其它两地的车辆情况并可进行预约登记。假设系统采用分布式数据库, 在每地均有自己的数据库系统, 用户预约和租借本地车辆仅限于在本地数据库系统内操作, 而当预约其它地方的车辆时则需要涉及到异地远程数据库系统的查询和更新操作。并假设对异地存在预约需求时, 所有的预约情况记录都存于异地。

因此, 设地域集 $L = \{A, B, C\}$ 其中, A 表示北京, B 表示广州, C 表示上海

1.2 功能定义

系统的功能定义如图 2:

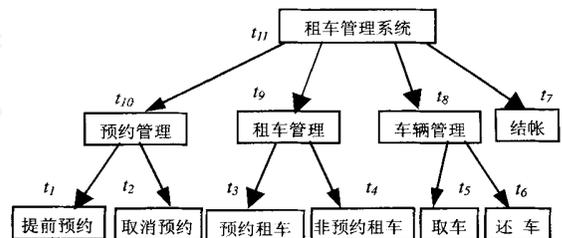


图 2 例 1 系统功能分解图

根据系统功能分解图, 可以得到系统底层功能集 $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$ 。

1.3 资源需求分析

1.3.1 系统信息资源集定义

运行所需的数据资源主要有

- r_1 —— 预约单记录
- r_2 —— 租借合约记录
- r_3 —— 客户租借情况登记
- r_4 —— 车辆记录
- r_5 —— 客户付款记录 即系统数据资源集为

$$R_d = \{r_1, r_2, r_3, r_4, r_5\}$$

运行应用所需的各客户端 CPU 资源集 $R_{cc} = \{cca, ccb, ccc\}$, 不同的元素代表不同地域的客户机;

各地域运行所需的网络资源 $R_n = \{nca, ncb, ncc\}$ 。

1.3.2 功能集上的数据资源约束集合定义

设例 1 功能集 T 上的资源约束集合为 P_T , 则有

$$P_T = \{P_{t1}, P_{t2}, P_{t3}, P_{t4}, P_{t5}, P_{t6}, P_{t7}\}$$

其中 $P_{\alpha 1} = \{r_1, r_4\}, P_{\alpha 2} = \{r_1, r_4\}$

$P_{\beta 3} = \{r_1, r_2, r_3, r_4\}$

$P_{\beta 4} = \{r_2, r_3, r_4\}, P_{\beta 5} = \{r_2\}$

$P_{\beta 6} = \{r_2\}, P_{\beta 7} = \{r_2, r_4, r_5\}$

分别为功能集合 T 中各基本功能结点上的资源约束子集

1.4 执行图

为研究系统的动态行为,有必要了解系统各事件(或称为功能)的执行顺序和流程,因此可以采用执行图来表示其执行过程 而对于在事件间有延迟或概率选择的情况则可在图中相关的弧上标出 下面给出执行图的规范定义:

定义 执行图 EG 是一个五元组 $EG = \langle E, X; F, A, W \rangle$

其中 $E = \{e\}$ 表示系统入口, $X = \{x\}$ 表示系统出口,两者均只是系统入出的标识

$F = \{F_R, F_S\}$ 代表系统的功能集合

其中 F_R 是入口模块集合,表示第一次进入系统可执行的操作;

$F_R \cap F_S = \emptyset, F_R \cup F_S = F;$

$F \cap X = \emptyset.$

A 表示功能间的连接关系,是有序偶 f_1, f_2 的集合,满足:

$A \subseteq \{E \times F_R\} \cup \{F_R \times F_S\} \cup \{F_S \times F_S\} \cup \{F_R \times X\} \cup \{F_S \times X\};$

$Dom(A) \cap Cod(A) = E \cup X \cap F;$

其中

$Dom(A) = \{x \in E \cup F \mid \exists y \in X \cup F, (x, y) \in A\}$

$Cod(A) = \{y \in X \cup F \mid \exists x \in E \cup F, (x, y) \in A\}$

有序偶 f_1, f_2 中 f_1 称为前功能, f_2 称为后功能

$W : A \rightarrow \{d, [g], [q]\}$ 其中:

d —— 从一项功能到另一项功能的延迟时间,且该延迟时间可能是一个服从于某一分布函数的变量

g —— 判断条件,值为布尔型 True 或 False,缺省为 True

q —— 从一项功能到另一项功能的概率, $q \in [0, 1]$

符号 $[\]$ 表示该项可选

如 $f_1, f_2 \in \{1, Z > 0, 0.5\}$ 表示执行功能 f_1 后若 $Z > 0$ 则将以 50% 的概率在 1 个时间单位的延迟后执行 f_2

对于 $f_1, f_2 \in \{E \times F_R\} \cup \{F_R \times X\} \cup \{F_S \times X\}$

则总有

$W : f_1, f_2 \in \{d\}$

一般来说, g, q 只在出现功能分支时才赋值

根据上述定义在考察系统后,可以得到本系统的执行图如图 3,在图 3 中凡无 g, q ,且 $d = 0$ 的弧上的 W 均未标出

1.5 执行图到 PetriNet(PN) 的转换

PN 网 N 是一个六元组 $N = \{P, T; H, R, D, G\}$,其中:

P 是位置集合, T 是变迁集合, H 是有向弧集合;

$R : T \rightarrow \{q\}$ 将变迁与概率分布相关联;

$D : T \rightarrow \{d\}$ 将变迁与延迟时间分布相关联;

$G : T \rightarrow \{g\}$ 将变迁与条件判断相关联

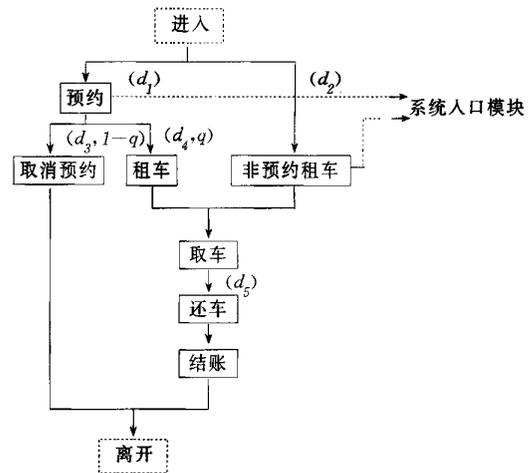


图 3 租车管理系统执行图

执行图 EG 与一个 PN 网 $N = \{P, T; H, R, D, G\}$ 相对应, EG 中 F 的每一个元素 f 均可转换为 N 中的一个变迁 t , 设 $F_R \rightarrow T_R, F_S \rightarrow T_S$. EG 中的 A, W 的转换方式如下:

1) 系统入口模块 F_R 的表示: F_R 中的功能模块代表用户第一次进入系统时执行的功能操作,其执行策略方式及用户到达情况服从于某一特定分布,在 N 中必须通过添加前置位置和前置变迁来表示用户进入系统的过程

规则 1 设执行图 EG 中 $f \in F_R$, 其对应的变迁为 $t \in T_R$ 则为了正确表示系统入口模块 f 的激发情况, 必须在变迁 t 前增加前置位置 p_R 和前置变迁 t' , t' 表示客户进入系统并激活功能 t , 该变迁的激发与服从某特定分布的随机时间相关联

$$\begin{aligned} \text{从而使得 } N: H &= H \quad \{t' \times p_R\} \quad \{p_R \times t\} \\ P &= P \quad \{p_R\} \\ T &= T \quad \{t'\} \end{aligned}$$

同时将 $A \subseteq \{E \times F_R\}$ 中对应的 W 添加至网 N 的 R 中, 即

$$W: A \quad \{d\} (EG) \Rightarrow R: t' \quad \{d\} (N)$$

因此, “预约”功能和“非预约租车”功能可以分别转换为图 4:

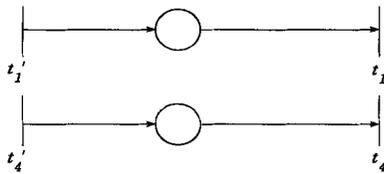


图 4 转换图例 1

2) 系统内部功能联系的转换: 此处功能联系指在 F_R 与 F_S 和 F_S 与 F_S 之间的执行过程关系, 即对于 EG 中 $A \subseteq \{F_R \times F_S\} \cup \{F_S \times F_S\}$ 中的有序偶的转换, 可以分为两大类来进行考虑: a) A_1 满足 $W: A_1 \quad \{0\}$ 即操作之间的执行不存在延迟; b) A_2 满足 $W: A_2 \quad \{d\}$ 且 $d > 0$ 或服从于某一分布即操作之间的执行存在延迟

A_1 无延迟: 功能的执行间隔可以忽略为 0, 即

$$W: f_1, f_2 \quad \{0\}$$

规则 2 设执行图 EG 中, $f_1, f_2 \in F$, $f_1, f_2 \in \{F_R \times F_S\} \cup \{F_S \times F_S\}$, $W: f_1, f_2 \quad \{0, [g, q]\}$, f_1, f_2 在网 N 中的对应变迁分别为 t_1, t_2 , 则 t_1, t_2 间的连接可直接通过在变迁间添加一个位置节点 p_{12} 来进行转换

$$\begin{aligned} \text{从而使得 } N: H &= H \quad \{t_1 \times p_{12}\} \quad \{p_{12} \times t_2\} \\ P &= P \quad \{p_{12}\} \end{aligned}$$

同时将 A_1 对应的 W 函数添加到网 N 中:

$$W: A_1 \quad \{d, [g], [q]\} (EG)$$

$$\Rightarrow D: t_2 \quad \{0\} (N)$$

$$G: t_2 \quad \{g\} \quad (\text{若存在})$$

$$R: t_2 \quad \{q\} \quad (\text{若存在})$$

如

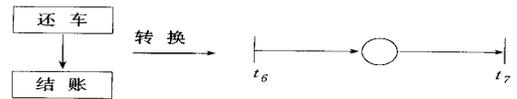


图 5 转换图例 2

A_2 有延迟: 功能之间存在延迟过程, 则不能再利用直接增加位置节点的方式进行转换, 否则无法表示其延迟特性

规则 3 设执行图 EG 中, $f_1, f_2 \in F$, $f_1, f_2 \in \{F_R \times F_S\} \cup \{F_S \times F_S\}$, $W: f_1, f_2 \quad \{d, [g, q]\}$ ($d > 0$), f_1, f_2 在网 N 中的对应变迁分别为 t_1, t_2 , 则 t_1, t_2 间的连接可以通过增加前变迁后置位置节点 p_1 、中间延迟变迁 t_{12} 、后变迁前置位置节点 p_2 来进行关联关系的转换:

$$\begin{aligned} \text{从而使得 } N: H &= H \quad \{t_1 \times p_1\} \quad \{p_1 \times t_{12}\} \\ &\quad \{t_{12} \times p_2\} \quad \{p_2 \times t_2\} \\ P &= P \quad \{p_1, p_2\} \\ T &= T \quad \{t_{12}\} \end{aligned}$$

同时将 A_2 对应的 W 函数添加到网 N 中:

$$W: A_2 \quad \{d, [g], [q]\} (EG)$$

$$\Rightarrow D: t_{12} \quad \{d\} (N)$$

$$G: t_{12} \quad \{g\} \quad (\text{若存在})$$

$$R: t_{12} \quad \{q\} \quad (\text{若存在})$$

如从“取车”到“还车”的过程可以用图形表示为

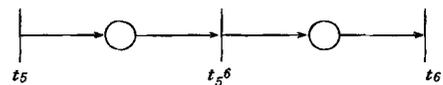


图 6 转换图例 3

3) 功能映射关系的合并: 经过以上步骤后, EG 中所有的功能和功能间对应关系已经转换到网 N 中, 但由于分支功能和归结功能的存在, 使得网的规模过大, 出现了不必要的重复, 因此可以对这些转换后的位置或变迁节点进行合并, 以减少模型的规模

分支功能的表示: 在执行图 EG 的功能元组中, 具有相同前功能元素的两个或多个后功能元素构成了共享前功能元素的后续分支功能。它能表示在执行完某一功能后有选择地执行某些功能, 体现并行过程的触发及条件判断后不同功能的执行情况。对它们的合并策略是将在功能元组间添加的前功能后置位置节点进行合并, 同时在弧上标出不同功能选择的判断条件和概率

策略 1 在网 N 中, 若对于变迁 $t \in T$, 存在 $\{ t, p_1, t, p_2, \dots, t, p_n \} \subseteq H (n > 1)$

则可添加一个新位置节点 p' , 同时将上述弧合并为 $\{ t, p' \}$, 而原始 N 中元组 $p_i, t_i (i = 1, \dots, n)$ 可转化为 $p', t_i (i = 1, \dots, n)$

即 $P = (P - \{p_1, p_2, \dots, p_n\}) \cup \{p'\}$

$H = (H - \{ t, p_1, p_1, t_1, t, p_2, p_2, t_2, \dots, t, p_n, p_n, t_n \}) \cup \{ t, p', p', t_1, p', t_2, \dots, p', t_n \} (i = 1, \dots, n)$

通过合并能够减少 $n - 1$ 条弧和 $n - 1$ 个位置

根据上述合并策略, 从“预约”开始的两种功能可以转换为

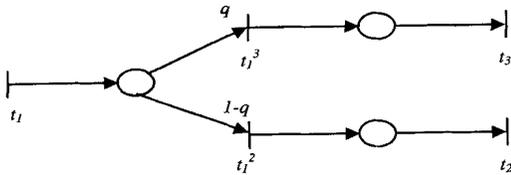


图 7 转换图例 4

归结功能的表示: 在执行图 EG 的功能元组中, 两个或多个具有相同后功能元素的功能元组的共享后功能构成了所有前功能元素的归结

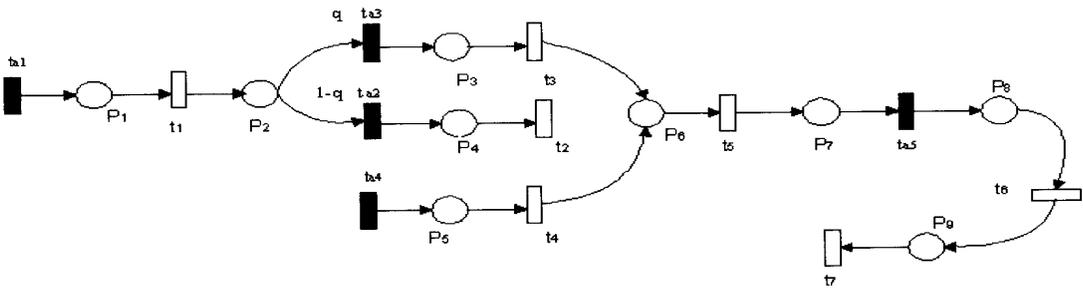


图 9 应用层模型

分支功能与归结功能的统一: 在执行图 EG 中, 归结功能可能是某一个前功能的后续分支功能, 这时必须进行特殊处理

策略 3 在执行图 EG 中, 若 $A_1 = \{ f_1, f_2, f_3, \dots, f_n, f \} \subseteq A (n > 1)$, 同时有 $A_2 = \{ f_i, f_{i1}, f_i, f_{i2}, \dots, f_i, f_{im} \} \subseteq A (n > i > 1, m > 0)$. 则 f 既是 f_1, f_2, \dots, f_n 的归结功能又是 f_i 的分支功能之一, 此时 A_1, A_2 的转换方式将有所变化, 其步骤如下:

按策略 1 进行集合 A_2 的转换;

功能 它表示在执行不同的功能后均要执行的共有功能操作 对它们的合并策略是将在功能元组间添加的后功能前置位置节点进行合并

策略 2 在网 N 中, 若对于变迁 $t \in T$, 存在 $\{ p_1, t, p_2, t, \dots, p_n, t \} \subseteq H (n > 1)$

则可添加一个新位置节点 p' , 同时将上述弧合并为 $\{ p', t \}$, 而原始 N 中元组 $t_i, p_i (i = 1, \dots, n)$ 可转化为 $t_i, p' (i = 1, \dots, n)$

即 $P = (P - \{p_1, p_2, \dots, p_n\}) \cup \{p'\}$

$H = (H - \{ t_1, p_1, p_1, t, t_2, p_2, p_2, t, \dots, t_n, p_n, p_n, t \}) \cup \{ p', t, t_1, p', t_2, p', \dots, t_n, p' \} (i = 1, \dots, n)$

通过合并能够减少 $n - 1$ 条弧和 $n - 1$ 个位置

根据上述原则, 从“预约租车”和“非预约租车”到“取车”的过程可表示如下:

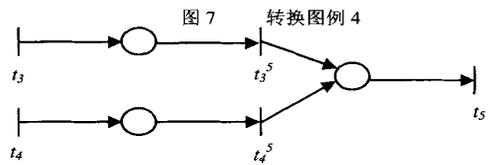


图 8 转换图例 5

对除 f_i, f 外的集合 A_1 中的元素按策略 2 进行转换, 并得到弧 p_1, t (p_1 为转换过程中新增加的位置);

若 f_i, f 之间存在时间延迟, 即 $w: f_i, f \{d, [g], [q]\} (d > 0)$, 则按策略 2 进行归结功能的表征;

若 f_i, f 之间不存在时间延迟, 即 $w: f_i, f \{0, [g], [q]\}$. 设网 N 中, $f_i \rightarrow t_i, f \rightarrow t, A_2$ 在转换过程增加的位置节点为 p_2 , 则 t_i, p_2, p_2, t 表明了 EG 中的关系 f_i, f , 此时

需在 p_2, t 之间添加一个变迁节点 t' 和位置节点 p' , 使得

$$P = P \cup \{p'\}$$

$$T = T \cup \{t'\}$$

$$H = (H - \{p_2, t\}) \cup \{p_2, t', t', p', p', t'\}$$

然后按照策略 2 进行 p', t' 的合并表征

1.6 生成系统顶层模型——应用层模型

利用上述转换规则和合并策略就可以得到系统应用层模型(图 9):

在网 N 中的 T 分为两部分: $T = \{T_o, T_a\}$

其中 T_o 对应由 EG 中的功能模块 F 转换而来的变迁, 并由下级子网来细化; T_a 对应转换过程中增加的辅助变迁, 它的激活时间与某变量相关

因此, 对于本系统而言:

$$T_a = \{ta_1, ta_2, ta_3, ta_4, ta_5\}$$

$$T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$$

在得到应用层模型后, 根据仿真工具 Design/CPN 的实际使用可以得到修改后的应用层模型图 10, 两者之间在结构上没有差异, 只是为了保证模型的正确运行而增加了一些位置和变迁节点

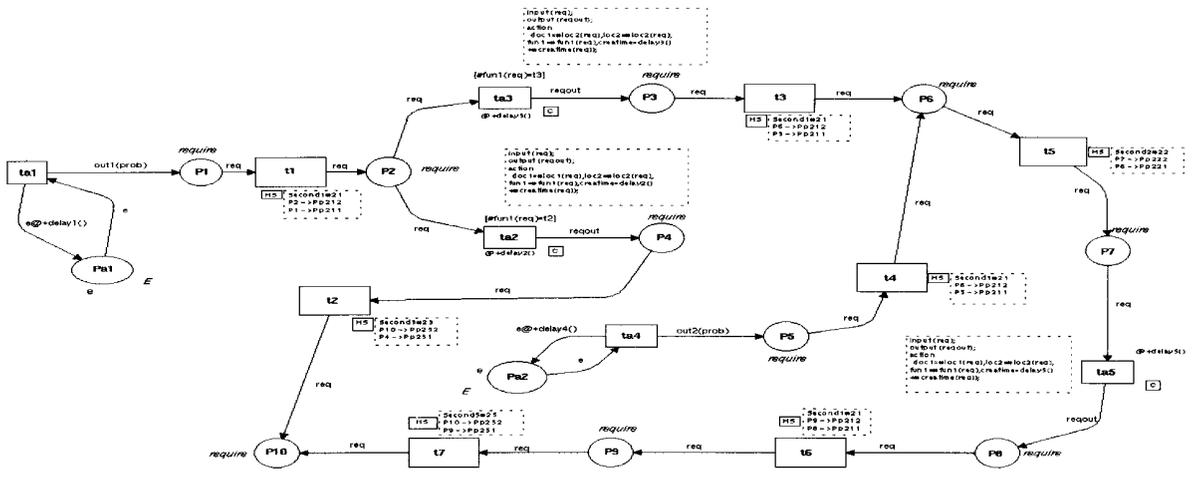


图 10 修改后的应用层模型

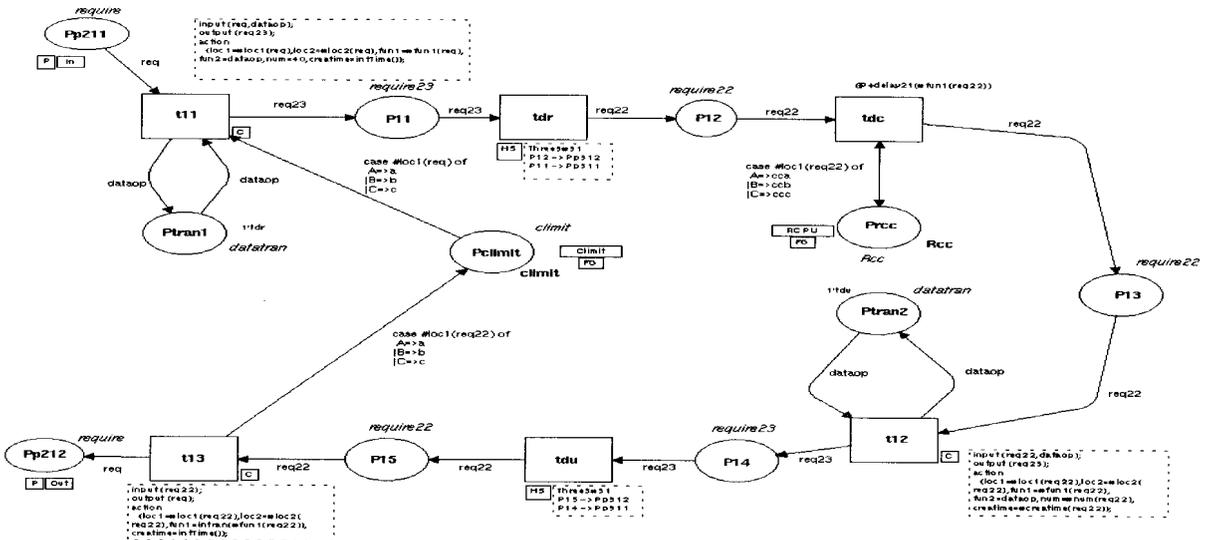


图 11 数据管理层 HS-Rou

1.9 应用前景分析

在系统模型建立后, 下一步工作则是要模拟系统的运行情况, 观测系统行为特性, 评价系统性能状况, 并对系统资源配置方案进行调整和优化。文[12]以文中模型为基础, 建立了信息系统的评价指标体系, 并对与例1相对应的系统动态模型进行仿真运行, 得到了系统响应时间、平均等待时间和平均排队队长等量化指标, 进而可以评价系统动态性能状况并对系统配置方案进行选择和调整。同时, 层次CPN模型所具有的下述优势保证了企业复杂系统的建模工作可以顺利进行:

通过建立系统执行图来生成系统动态模型, 可以保证整个建模过程的规范化, 并实现部分建模工作的自动完成;

CPN模型部件具有可重用性, 能够减少系统建模的工作量;

层次模型提供了更好的可扩展性, 建模者可以根据需要方便地扩展模型

因此, 系统开发者可以以提出的CPN模型为基础来对企业信息系统进行建模, 进而分析系统性能状况并调整系统资源配置方案

2 结束语

在系统顶层设计阶段处理好信息资源配置问题对信息系统是至关重要的, 这一问题的解决与否直接影响到系统的效率和性能, 并关系到信息系统的生存和发展。作者提出了一种基于CPN的信息系统资源配置模型的建立方法, 给出了执行图的规范化定义和执行图到Petri网的转换方法。通过文中定义的规范化步骤, 开发者可以对信息系统进行建模分析, 并在综合考虑各种因素后达到资源配置的合理化和优化。该方法的实际意义:

提供了一个对信息系统资源配置情况进行建模分析的有效手段;

建模过程是规范化的, 并具有自动完成功能, 从而使得对复杂大系统的建模成为可能;

可以通过仿真方法对模型进行性能分析和评价;

可以指导系统建设过程中的信息资源配置过程

文中提出的方法是对信息系统资源配置分析的一种尝试性方案, 其理论体系和实施方案尚需进一步分析研究

参考文献

- [1] 孟广均等著. 信息资源管理导论[M]. 北京: 科学出版社, 1998
- [2] 李文波, 吴冲锋, 王意冈. Petri网分析企业组织效率的探讨[J]. 管理科学学报, 1999, 2(2): 49-56
- [3] Gan Renchu, Yan Zhijun, Wang Jingguang. Modeling and evaluation of resources allocation for information systems based on Petri nets[C]. The 2nd Asia-Pacific Conference on Industrial Engineering and Management System; Knazawa, Japan, 1999. 171-176
- [4] Daneva M. The role of benchmarking in information system development[C]. Advanced Information Systems Engineering, 1997. 421-430
- [5] Opdahl A L. Performance engineering during information system development[M]. 1996
- [6] Brataas G. Performance engineering method for workflow systems: An integrated view of human and computerised work processes[D]. Norwegian University of Science and Technology (NTNU), 1996
- [7] Ferscha A. A Petri net approach for performance oriented parallel programming design[J]. Journal of Parallel and Distributed Computing, 1992, 15(3): 188-206
- [8] Jensen K. Coloured Petri nets: basic concepts, analysis methods and practical use Vol 1: Basic Concepts[M]. Berlin: Springer-Verlag, 1992
- [9] Design/CPN Reference Manual Version 2.0[M]. Meta Software Corporation, 1993
- [10] Paulson C L. ML for the working programmer[M]. 2nd Edition. Cambridge: Cambridge University Press, 1996
- [11] Cherkasova L, Kotov V, Rokicki T. On scalable net modeling of OLTP. In PNPM 93: Petri Nets and Performance Models[C]. Proceedings of the 5th International Workshop, 1993. 270-279
- [12] 颜志军, 甘初初. 信息系统动态性能评价指标体系的建立和分析[C]. 2000年管理科学与信息系统学术会议论文

集, 2000, 186-190

- [13] Slooten van Kees Contingent information systems development[J]. J. Systems Software, 1996, 10(2): 150-162
- [14] Heuser C A. Constructs for modeling information systems with Petri nets[C]. Application and Theory of Petri Nets, 1992 225-241

Dynamical modeling of resource allocation for information system based on coloured Petri nets

YAN Zhi-jun, GAN Ren-chu

School of Management and Economics, Beijing Institute of Technology, Beijing 100081, China

Abstract This work is devoted to a dynamical modeling method of resource allocation for information systems based on coloured Petri nets. The method can be used in the more complex enterprise information systems. The definitions of system hierarchy model and execution graph are proposed. System developers can pay more attention on different level of the model according to their concern. And the transformation method between execution graph and petri net are analyzed. Based on the proposed dynamical hierarchy model, system developers can evaluate the system dynamical performance and adjust the resource allocation of information systems to meet the performance requirement. The modeling process of information systems is given in detail. And application example is presented.

Key words: information system; coloured Petri net; resource allocation; performance evaluation