

随机 Petri 网性能计算软件关键技术的研究

丁 静, 韩 强

(北方民族大学 计算机科学与工程学院, 宁夏 银川 750021)

摘 要:性能计算对随机 Petri 网模型的性能分析具有重要作用。以研究随机 Petri 网性能计算软件关键技术为主要目的, 提出了一种独立于计算(CIM)的性能计算软件体系结构元模型, 并以此元模型为依据, 遵循面向对象的系统分析与设计方法给出该软件独立于平台(PIM)的软件体系结构模型, 对核心算法设计方案采用先计算基础性性能数据, 再计算扩展性能数据的策略分步骤完成, 并给出了基于 MVC 模式实现的 PSM 模型实例和一个电子商务应用案例。最后, 给出了同类性能计算软件的设计可适用此开发方法的初步结论。

关键词:随机 Petri 网; 元模型; 马尔可夫链; 软件体系结构; 电子商务

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)11-0057-04

Study of Visualized Performance Computation Software Models for Stochastic Petri Net

DING Jing, HAN Qiang

(School of Computer Science and Engineering, North University for Ethnic, Yinchuan 750021, China)

Abstract: Performance computation has significant meaning to performance analysis of models based on stochastic Petri net (SPN). To aim at research on the key technology of performance computation software for the models, firstly, give a compute-independence metamodel (CIM) for the software-architecture based on SPN, and according to it, give a platform-independence model (PIM) through methods of oriented-object. Promoted a design project of firstly calculating foundation performance data and again calculating expanded performance data. Finally, give a platform-specified model case based on MVC and an electronic-agriculture services case. In the end, a conclusion that similar software can adopt the method is made.

Key words: stochastic petri net; metamodel; Markovchain; software-architecture; electronic-agriculture services

0 引 言

20 世纪 80 年代初, Molloy 提出了在 Petri 网^[1]基础上, 即在每个变迁的可实施与实施之间联系一个随机的延迟时间, 形成随机 Petri 网。

该文分别以文献[2]中 ABC 方法使用的模型元素建立独立于计算(Compute-Independence Model, CIM)的元模型, 以文献[3]中统一建模语言 UML 为表示工具, 将 CIM 模型转化为独立于平台(Platform-Independence Model, PIM^[4])的软件体系结构, 进而利用类图和核心算法流程等元素对 PIM 模型进行精化, 介绍了该软件的设计方案。给出一个基于 MVC(Model-View-Controller)模式和特定平台(Platform-Specified

Model, PSM^[4])的软件实例。

1 软件设计方案

一个可视化随机 Petri 网性能计算软件的基本功能包括:

- (1) 模型输入、修改、存储和读取;
- (2) 模型基础性性能数据的计算;
- (3) 模型扩展性能数据参数的输入;
- (4) 模型扩展性能数据的计算;
- (5) 模型性能数据的输出;
- (6) 模型的输出。

根据上述抽象功能, 可看到该软件系统应该由 6 个相对独立的构件组成, 可建立一个独立于随机 Petri 网计算的软件体系结构 CIM 元模型, 如图 1 所示。

依据图 1 所示 CIM 元模型, 结合随机 Petri 网性能计算理论, 以 UML 语言^[5]为设计工具, 参考文献[2]介绍的 ABC/ADL 到 UML 的映射规则, 即构件和

收稿日期: 2009-03-04; 修回日期: 2009-06-27

基金项目: 国家科技支撑计划课题项目(2007BAD33B03); 宁夏高等学校科学研究项目(2008JY009); 宁夏自然科学基金(NZ0955)

作者简介: 丁 静(1976-), 女(回族), 硕士, 研究方向为 Petri 网理论及应用、软件工程、嵌入式软件。

复杂连接器映射为包、功能规约映射为接口、接入点映射为抽象类以及构件内部规约映射为说明的指导原则,给出独立于实现平台的 PIM 模型^[4],见图 2。

从图 2 的 PIM 模型中,可以看到该模型的核心细节涉及两点:随机 Petri 网类的组成和基础性能数据计算算法。其中随机 Petri 网类的设计是该软件的数据基础,而基础性能数据计算算法是该软件的计算基础,二

者共同组成了该软件运行的核心轴线。使用者可通过扩展性能数据的计算结果反馈调整随机 Petri 网模型,进而形成该软件运行的总路线,如图 3 所示。

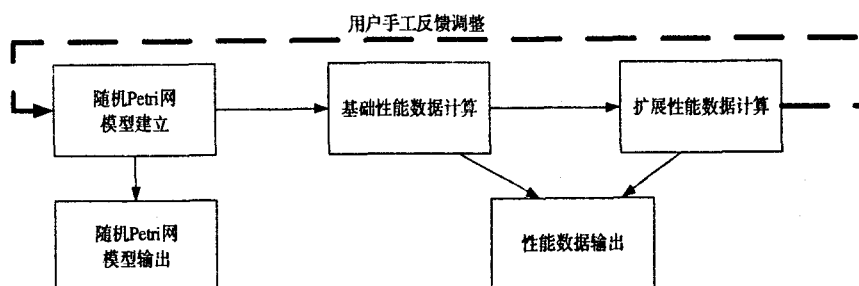


图 3 软件运行路线图

根据上述分析,给出该软件数据基础和计算基础这两个核心描述:

(1) 数据基础:根据随机 Petri 网定义,任何一个随机 Petri 网模型应由 3 类主要元素组成,即节点(库所)、变迁和弧。由此建立了以 BasicEntity 为基类、以 Entity 为泛化基类、以 Node 和 Transition 为普通类、以 Arch 为独立类为结构的随机 Petri 网模型类图,如图 4 所示。

(2) 计算基础:根据随机 Petri 网性能计算公式,任何一个随机 Petri 网模型与一个马尔可夫链同构,即对于一个具有 N 个状态标识的随机 Petri 网模型,按照马尔可夫随机过程理论,可将该马尔可夫链转换为一个由 $N+1$ 个等式组成的 N 元方程组,进而选择某种算法(例如高斯消元法)解该方程组获得各状态标识的稳定状态概率,以此稳定状态概率作为基础性能数据,计算扩展性能数据。

下面给出基础性能数据计算算法。

1) 遍历 SPN 模型的每一个节点 Node,根据当前 Node 的 Token 数更新当前标识。若遍历完毕,转 5)。

2) 遍历 SPN 模型的每一个变迁 Transition,若遍历完毕,转 1)。

3) 遍历 SPN 模型的每一个弧 Arch,若遍历完毕,转 2)。

4) 根据弧的 beginEntity, endEntity 和当前 Node, Transition 设置节点变迁邻接矩阵或变迁节点邻接矩阵。转 3)。

5) 将当前标识插入标识数组。

6) 遍历标识数组每一个未扩展标识,若所有标识均已扩展,转 13)。

7) 遍历每一个变迁,若遍历完毕,转 6)。

8) 若当前标识在当前变迁下可实施,

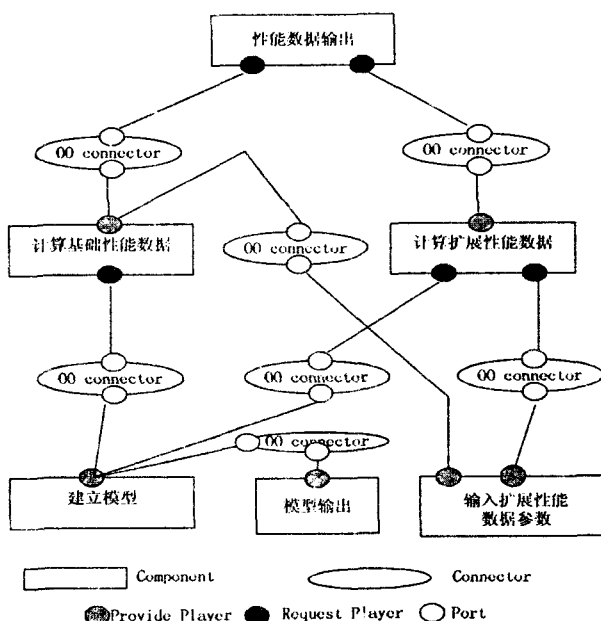


图 1 软件体系结构 CIM 元模型

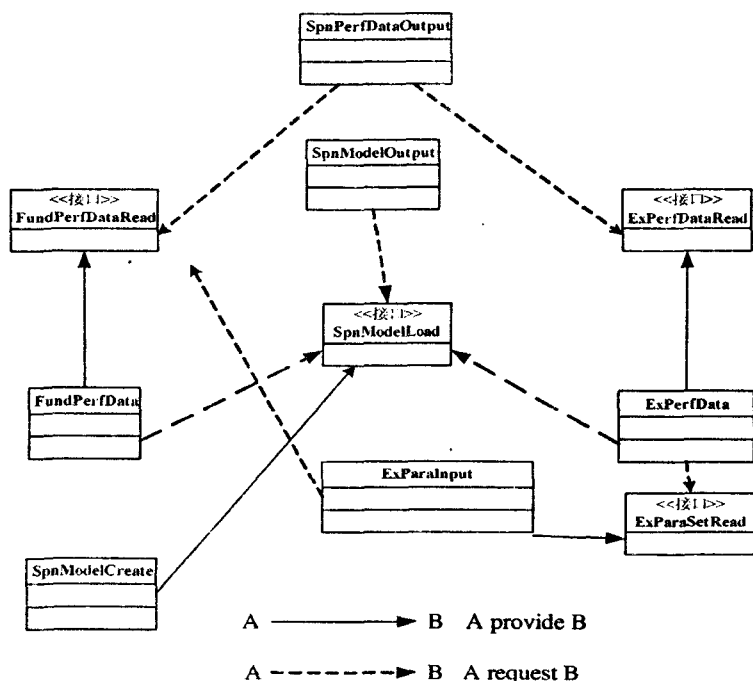


图 2 软件体系结构 PIM 模型

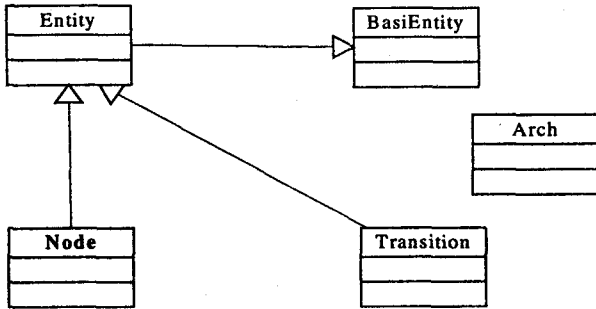


图 4 软件核心类图

即满足:标识各节点 token 数 > 该节点当前变迁邻接数,并且标识各节点容量 ≤ 当前标识该节点 token 数 - 该节点当前变迁邻接数 + 当前变迁该节点邻接数,则进入下一步,否则,转 7)。

9) 创建一个标识。若该标识存在一个节点容量大于该 SPN 模型该节点最大容量,则转 12)。

10) 该标识是否是新标识,若是则置标识序列号加 1,插入标识数组;否则,置标识序列号为标识数组中原标识序列号。

11) 若当前标识与创建的标识已经在标识转移速率矩阵中,则置转移速率矩阵相应值为原值加当前变迁实施速率;否则,创建转移速率矩阵中新关系,置关系值为当前变迁实施速率。转 7)。

12) 该 SPN 模型无界,算法失败,转 14)。

13) 该 SPN 模型有界,标识转移速率矩阵建立完毕,选择某种算法(例如高斯消元法)解该方程组获得各状态标识的稳定状态概率。算法成功。

14) 算法结束,退出。

2 基于 MVC 模式的软件实例及应用

MVC(Model - View - Controller)^[6-8]中模型是应用程序的核心。它将应用程序的数据结构和事务逻辑进行封装,并集中体现应用程序的状态;视图实现模块的可视化人机界面^[5],它是应用程序的外在表现。当模型发生改变时,视图就会得到通知,可以访问模型的

数据但是不能改变这些数据;控制器控制整个框架中各个组件的协调工作,对用户的输入做出反应。根据上述对 MVC 模式的工作原理分析,基于 MVC 模式为设计模型,给出基于 Windows XP 平台,使用 VC++ 6.0 为开发环境的 PSM^[4]模型及其软件实例。

依据第 1 节中的 CIM 和 PIM 模型,可将软件需求和类集划分为三个部分:

1) 实现随机 Petri 网模型的建立、性能数据的输出、扩展性能参数的输入。

2) 随机 Petri 网模型性能数据的计算。

3) 控制协调(该部分属于软件非功能需求)。

上述三个部分分别对应 MVC 模式中的 View、Model 和 Controller。由此,PSM 模型如图 5 所示。依据该 PSM 模型进行软件开发获得软件实例 Petri 1.0。

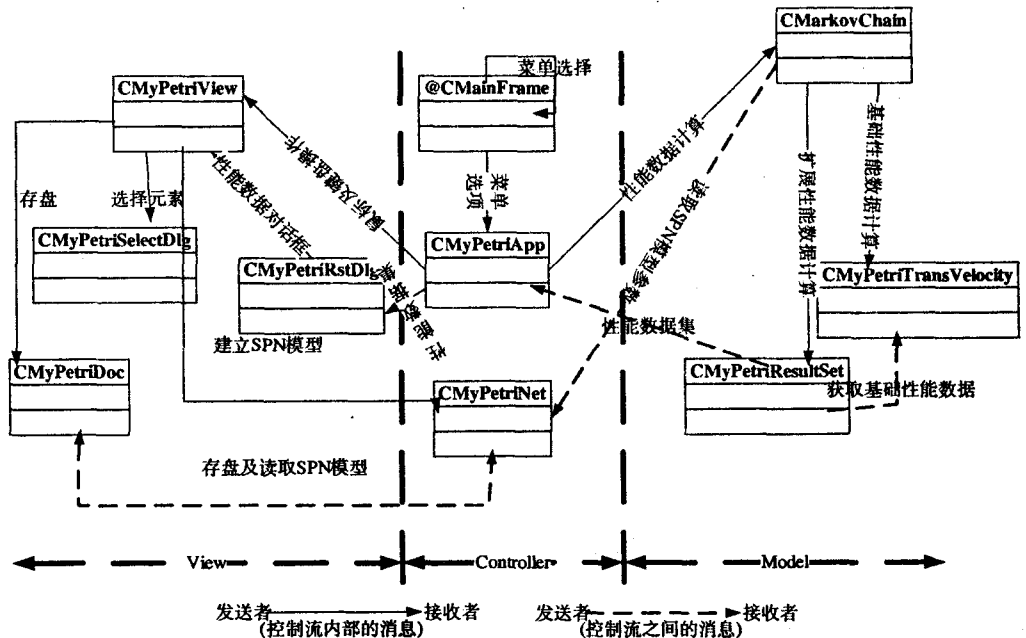


图 5 基于 MVC 的 PSM 模型

下面利用 Petri 1.0 基于电子农务平台的一个奶牛体征数据采集的应用案例。电子农务平台是信息化农务应用系统的基础,在本案例中 SPN 模型如图 6 所示,其中 P0 是“活动点”节点(或称库所),当 P0 包含标记时,表示相应的奶牛在自己的采集点内活动。T0 (T1)是“结束自我活动,进入体温(体重)待采集队列”变迁,当此变迁实施时,一只奶牛结束自我活动,进入 P1(P2)采集点等待队列节点。T2(T3)是“结束等待,进入体温(体重)采集队列”变迁,当此变迁实施时,一只奶牛结束等待活动,进入 P3(P4)采集点节点。P5 (P6)中包含标记时代表采集设备已准备好,P8 代表数据上传设备已准备好。T4(T5)是“结束采集活动,回到活动点节点”变迁。初始参数设置 T0(T1)转移速率为 1 只/小时;T2(T3)为 10 只/小时;T4(T5)为 2 只/

小时。根据图 6 所示 SPN 模型,该软件计算的性能数据结果是:该 SPN 模型共 12 个标识状态,例如标识 0 为“2 0 0 0 1 1 1”,稳定状态概率是 0.2590。节点 P0 的平均标识数是 0.88。

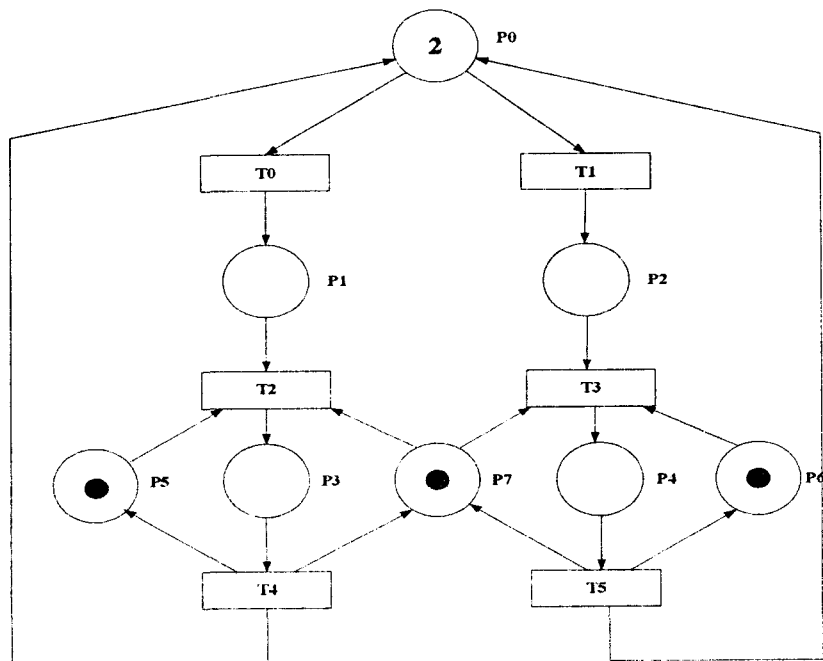


图 6 奶牛体征采集 SPN 模型图

3 结束语

通过研究可视化随机 Petri 网性能计算软件模型,分别建立了该软件的 CIM, PIM, PSM 模型,最后通过一个电子农务平台应用案例,给出了软件运行实例。由于采用了独立于随机 Petri 网的软件设计方法,而且

(上接第 56 页)

提供的内存泄漏与越界检测也简化了软件的开发流程,增强了软件的稳定性。

参考文献:

- [1] 王力生,仇志付,唐军敏. 嵌入式操作系统通用硬件层的设计[J/OL]. 单片机与嵌入式系统应用, 2006-10. http://www.mesnet.com.cn/html/magazine_view.asp?id=2029.
- [2] Serpanos D N, Karakostas P. Efficient Memory Management for High-Speed ATM Systems[J/OL]. Design Automation for Embedded Systems, 2004-11. http://www.library.utoronto.ca/pdf_test/kl08132001/354145.pdf.
- [3] 李小文. TD-SCDMA 第三代移动通信系统、信令及实现[M]. 北京:人民邮电出版社, 2005.
- [4] 斯托林斯. 操作系统——精髓与设计原理[M]. 第 5 版. 北京:电子工业出版社, 2006.
- [5] 黄贤英,王越,陈媛. 嵌入式实时系统内存管理策略

该性能计算功能是同类性能计算软件的基础功能,因此同类软件的设计亦可适用此开发方法。下一步的研究方向是如何解决原始输入参数的合法性验证等问题,以提高软件的可计算性。

参考文献:

- [1] Petri C A. Kommunikation mit automaten[M]. Bonn: Institut für Instrumentelle Mathematik, Schriften des IIM Nr. 3, 1962.
- [2] 梅宏,陈锋,冯耀东,等. ABC: 基于体系结构、面向构件的软件开发方法[J]. 软件学报, 2003, 14(4): 721-732.
- [3] OMG unified modeling language specification (Version 1.4) [S]. Needham: Object Management Group, Inc., 2001.
- [4] 邵维忠,杨芙清. 面向对象的系统分析[M]. 第 2 版. 北京:清华大学出版社, 2004.
- [5] 邵维忠,刘昕. 可视化编程环境下人机界面的面向对象设计[J]. 软件学报, 2002, 13(8): 1494-1499.
- [6] 杨健. 基于 MVC 的论坛网站的设计与实现[J]. 计算机技术与发展, 2006, 16(6): 81-83.
- [7] 徐婷婷,段凡丁. MVC 设计模式在 B/S 开发中的研究与应用[J]. 计算机技术与发展, 2007, 17(6): 119-121.
- [8] 隋永,周家纪. MVC 在 J2EE 框架中的应用研究[J]. 计算机技术与发展, 2006, 16(12): 235-238.
- [9] 李毅. Slab 内存分配策略与移植[J]. 计算机技术与发展, 2007, 17(10): 168-170.
- [10] Friedrich L F, Stankovic J. A Survey of Configurable, Component-based Operating System for Embedded Applications[J]. IEEE MICRO, 2001, 5(6): 54-68.
- [11] 董庆丰,黄迪明. 一种适用于嵌入式系统的动态内存管理技术[J]. 微型机与应用, 2004(8): 52-55.
- [12] Leeman M. Methodology for Refinement and Optimisation of Dynamic Memory Management for Embedded Systems in Multimedia Applications[J]. The Journal of VLSI Signal Processing, 2005, 40(3): 383-396.
- [13] 刘小军,李秀娟. 嵌入式操作系统 VxWorks 的内存管理技术研究[J]. 电子科技, 2008, 21(6): 62-65.
- [14] 陈洋. VxWorks 下的内存管理[J]. 计算机工程, 2007, 33(8): 94-96.