

基于随机 Petri 网的工作流仿真

许春霞, 姜 浩

(东南大学 计算机科学与工程学院, 江苏 南京 210096)

摘 要: 工作流管理是近年来发展最为迅速的计算机应用技术之一, 被广泛应用于多个领域。Petri 网, 是对具有并发、同步、异步、冲突、资源共享以及不确定性等特点的离散事件系统进行建模的有效工具。针对工作流的性能评价问题设计和实现了随机 Petri 网仿真工具。介绍了工作流和随机 Petri 网的基本概念, 然后具体介绍了仿真策略和冲突解决方法的關鍵仿真技术问题, 最后结合具体实例对仿真工具的有效性和正确性进行验证。

关键词: 随机 Petri 网; 工作流; 性能评价; 仿真

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2009)04-0087-04

Workflow Simulation Based on Stochastic Petri Net

XU Chun-xia, JIANG Hao

(School of Computer Science & Engineering, Southeast University, Nanjing 210096, China)

Abstract: Workflow management has become one of most rapidly developed technology in computer science and has been widely used in a lot of fields. Petri net is an effective model tool for discrete event system with the characteristics, such as concurrency, asynchronism, collision, resource sharing and so on. Aiming at the performance evaluation of workflow, designs and implements simulation Tool. Some of the basic concepts about workflow and stochastic Petri net are introduced. The key simulation techniques of simulation strategy and conflict resolution policy are then presented. An example was illuminated to show the validity and correctness of the simulation tool.

Key words: stochastic Petri net; workflow; performance evaluation; simulation

0 引 言

工作流是对业务进程进行的形式化描述, 包括描述任务间的依赖关系的工作流逻辑和在此逻辑上增加显性内容的工作流语义。工作流管理是按照工作流逻辑次序依次检查已完成的任务是否符合标准并启动下一任务或根据工作流语义选择下一步任务并启动之^[1]。近年来, 工作流技术的研究和应用已引起了研究人员开发人员和用户的普遍关注, 被广泛应用于多个领域, 例如计算机支持的协同工作、无纸办公表单传递、协作系统和办公自动化等。

工作流的性能评价是工作流系统研究和应用的主要理论基础和支撑技术, 是工作流系统研究和应用的重要内容。性能评价是指对工作流管理系统的一些性能指标如平均吞吐量、平均等待时间、平均等待处理的工作项个数、资源占用率等进行计算并依此设计预测和改进原有系统, 提供有效的数学理论工具。目前已有的工作性能评价方法主要有: 马尔可夫分析方法、排

队论和仿真法。文中主要通过随机 Petri 网仿真工具来进行工作流性能评价。

1 工作流

工作流技术起源于生产组织和办公自动化领域。工作流的实质是利用计算机的通讯功能, 达到各个工作的通信、交流、协同工作等。使多个参与者之间按照某种预订的规则传递信息来实现业务目标。工作流就是一系列任务构成的集合, 包括任务间的通讯、合作、协调和同步关系。工作流管理系统 (Workflow Management System, WFMS) 是指支持多个组织进行异地、异步操作的软件工具, 它具有定义、执行、表示和控制工作流的多项功能。工作流管理系统通过在案例、任务和资源之间建立一种特定的联系, 使得所有任务按照案例的需求有次序的进行。工作流管理的最大优点是将应用逻辑和过程逻辑分开, 在不修改具体功能的情况下, 通过修改模型改变系统功能, 完成对生产经营部分过程或者全过程的集成管理, 可有效地把人、信息和应用工具合理地组织在一起, 发挥系统最大功能。

WFMS 提出: 复杂的工作流模型都可以用一组简单的结构复合而成^[2]。这四个基本结构是: 顺序结构、

收稿日期: 2008-07-20

作者简介: 许春霞 (1984-), 女, 江苏启东人, 硕士研究生, 研究方向为 Petri 网应用; 姜 浩, 副教授, 研究方向为工作流应用研究。

并行结构、选择结构、循环结构。这四种结构可以等效的用 Petri 网对应表示。如图 1 所示。

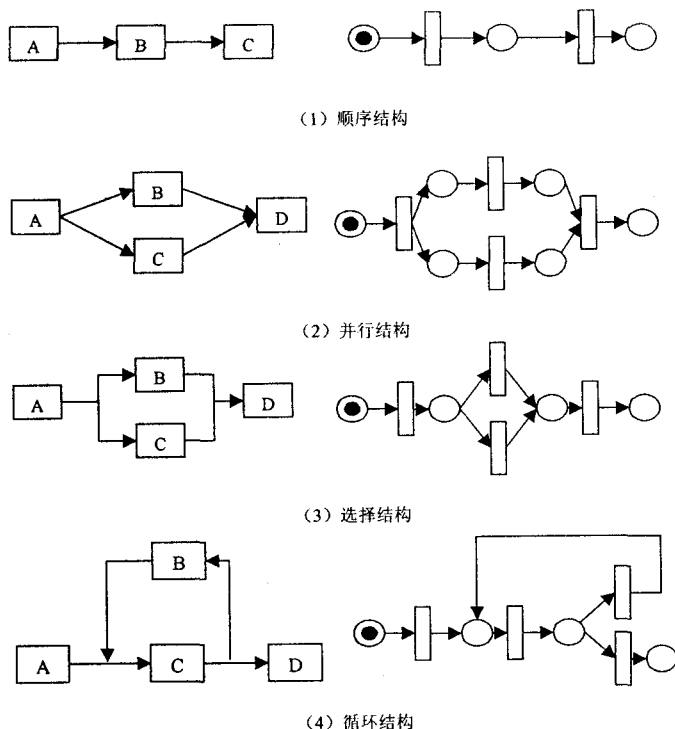


图 1 workflow 模型的四种基本结构

工作流性能主要反映工作流定量方面的特性,比如,案例的完成时间、单位时间内处理的案例数量、资源利用率和在预定的标准时间内完成案例的百分比等等。考察工作流的性能存在多种方法,下面介绍最常用的三种方法^[3]。

(1) 马尔可夫分析方法。

马尔可夫方法的主要研究对象是一个运行系统的状态和状态转移。对给定的工作流,可以自动生成一个马尔可夫链,利用它可以分析工作流的许多方面。通过成本和时间的引入扩展马尔可夫链,就能获取一系列性能指标。

(2) 排队论。

排队论是研究大量服务过程的数学理论。它是目前应用最广泛的一种系统分析和优化的方法。在工作流中,总有很多活动队列,它们在排队等待资源。如果对一个拥有大量的相同资源的队列感兴趣,就可以将系统看作一个单队列系统。而要评价整个工作流,就需要研究队列网络。

(3) 仿真。

工作流模型仿真是利用离散事件驱动的仿真引擎模拟执行工作流模型中的各项活动,自动推进工作流实例。经过多次仿真运行,得到一系列关于工作流模型运行的统计数据。用户可以在这些仿真统计数据的

基础上进一步分析和评价一些关键性能指标:如平均响应时间、平均等待时间、资源利用率、服务水平等。

2 随机 Petri 网

Petri 网的诞生源于西德计算机科学家 C. A. Petri 1962 年的博士论文,它是描述具有分布、并发、异步特征的系统有效工具。传统的 Petri 网一旦满足出发条件,就立即触发且发生持续时间为 0,没有任何延时就产生结果。而工作流是业务流程的全部或者部分自动化,是需要考虑时间性能指标,时间的引入非常重要,随机 Petri 网是非常适合描述系统时间性能指标的建模工具,它特别适合于时间性能指标的分析。

随机 Petri 网的定义如下:

用五元组 (P, T, F, M_0, λ) 来表示随机 Petri 网(Stochastic Petri Net, SPN),其中: P 是有限非空库所集; T 是有限非空变迁集; $P \cap T = \emptyset$; $F \subseteq P \times T \cup T \times P$ 是一个有向弧的集合; M_0 是 Petri 网的初始标识, M 是从 P 到非负整数 N 的映射, $\forall p \in P, M(p)$ 表示 M 标识库所下 p 的容量值。 $\lambda = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n)$ 是变迁平均发射速率集合, $1/\lambda_i$ 称为 t_i 的平均延迟时间。

随机工作流网的定义如下:

一个 Petri 网 $SWF = (P, T, F, M_0, \lambda)$ 是一个随机工作流网(Stochastic Workflow Net) 当且仅当:

- (1) SWF 是一个随机 Petri 网;
- (2) SWF 中存在两个特殊库所:库所 i 是开始库所(无输入),库所 o 是结束库所(无输出);
- (3) 对于 $\forall x \in P \cup T$, 存在一条从开始库所到结束库所得路径 C , 使得 x 是 C 上的一个节点;
- (4) 网的初始状态为 $M_0 = \{P_i\}$, 网的终止状态为 $M_0 = \{P_o\}$ 。

3 随机 Petri 网的仿真设计和实现

Petri 网建模工具主要包括三个部分:图形编辑器、网库管理和仿真模块。

图形编辑器是系统建模仿真的基础,它的作用是将系统的 Petri 网图绘制在窗口中。用户可以添加和修改库所,变迁和有向弧,并且进行参数设定。网库管理即是将用户画好的 Petri 网图以及图元的相关属性采用 Petri Net Markup Language(PNML)保存起来,使 Petri 网工具具有向另一种 Petri 网工具导出 Petri 网或者从其他的工具中导入 Petri 网的功能。PNML 是在 2000 年的 Petri 网应用和理论国际会议上提出的基于

XML 交换格式建议之一,其主要聚焦在如何解决由于不同的 Petri 网类型而导致的问题。

仿真模块是动态模拟 Petri 网模型在给定初始条件下 token 的流动及系统状态的变化,并将过程及结果记录下来,这是文中的核心内容,下面主要介绍仿真模块的设计实现。

3.1 基于事件调度的仿真策略

工作流模型仿真是利用离散事件驱动的仿真引擎模拟执行工作流模型中的各项活动,自动推进工作流实例。迄今为止,离散事件系统已经形成了三种基本的仿真策略,分别是事件调度(Event Scheduling, ES)、活动扫描(Activity Scanning, AS)和进程交互(Process Interaction, PI),其他仿真策略均是以这三种仿真策略为基础的。文中采用基于事件调度的仿真策略^[4]。

事件调度策略完整地描述系统中一个独立事件发生时所引发的各种变化及其操作步骤。其基本思想是:以事件例程作为仿真模型的基本仿真单元,按照事件发生的先后顺序不断执行相应的事件例程。在每一个确定发生时间的事件例程发生时,产生下一事件发生的时间,同时处理当前事件发生后对系统状态所产生的影响;条件事件不具有事件例程,对它的处理隐含在某一确定事件的事件例程中。因此,事件调度法所说的事件通常指确定事件。其仿真时钟推进机制是将仿真时钟推进到下一最早发生事件的发生时刻,体现了离散事件系统状态仅在离散时刻点上发生变化的特点。

3.2 冲突解决策略

在无时间的 Petri 网中,两个变迁的实施是完全不确定的,哪一种变迁的实施都有可能,一个变迁的实施使得另一个变迁不可实施。随机 Petri 网采用的是可触发持续时间(enabling duration)来表示时间,变迁在可触发后将在可触发持续时间内保持可触发状态,变迁的实施是瞬间完成的,即在同一时间移去输入托肯并产生输出托肯。时间延迟表现在强迫变迁在实施之前需要在指定的时间延迟内维持可触发状态。因此随机 Petri 网中,当有两个或两个以上变迁同时满足触发条件而共享库所中的 token 资源不能满足时冲突就会发生。变迁之间的冲突通常采用竞争机制

在随机 Petri 网中,有 3 种竞争策略可以采用:可以基于优先级机制,确定变迁的优先次序,具有高优先级的变迁可实施,而具有低优先级的变迁不可实施;给变迁不同的条件谓词,满足条件谓词的变迁先实施,不满足条件谓词的变迁不可实施;基于变迁实施时间;如果每一种变迁都有可能实施,任一变迁的实施使得其他变迁不可实施,则具有最短可触发持续时间的变迁具有最大的实施可能性。一般采用选择可触发持续时

间最短的变迁优先实施。当这个可触发持续时间最短的变迁实施之后,其他变迁的变迁使能被中断,其他变迁的可触发持续时间该如何修改仍需要讨论。

在随机 Petri 网中对每个时间变迁定义两个连续变量,一个叫时钟变量,用于表示变迁已经被使能的时间,只要变迁处于使能状态,则该变量值随时间同步增长;另一个叫抽样值变量,用于表示变迁的发射时间抽样值,即可触发持续时间。当时钟变量与抽样值变量数值相等,则变迁发射。变迁的使能被中断后,如何处理这两个变量就是所谓的变迁记忆策略(memory policy),目前常用的记忆策略有以下三种^[5]:

(1) 中断后再继续(preemptiver esume,采取此种记忆策略的变迁称为 prs 变迁):变迁使能状态的中断只是中止时钟变量的增长,对抽样值变量没有影响,抽样值变量的重新赋值(即重新抽样)和时钟变量的复位为 0 仅发生在变迁发射之后。这种记忆策略用于表示活动中断仅暂停执行现有的工作,活动重新开始后,继续完成被暂停的工作。

(2) 中断后以另一值开始(preemptive repeat different, prd 变迁):变迁的使能被中断后,时钟变量复位为 0,抽样值变量重新赋值,用于表示活动被中断后,完全抛弃已部分完成的工作。这种方法,几乎没有什么实际应用,因为这样就丢掉了采用 Petri 网建模的一个重要特性并发性,因为当一个事件发生后,其他事件的发生时间就要重新计算,这就不具有并发性。

(3) 中断后以相同值开始(preemptive repeat identical, pri 变迁):变迁的使能被中断后,时钟变量复位为 0,抽样值变量不变,其重新赋值仅发生在变迁发射之后。表示活动重新开始后,执行刚才被暂停的同一工作,但已完成的不算,而是重新开始。

3.3 仿真实现

根据上文的仿真策略和冲突解决策略,可以设计仿真算法。仿真的结论是对整个网运行的时间的一个评估。仿真流程如图 2 所示。

几点说明:

(1) 仿真开始前,所有变迁的可触发标志均为 false。扫描 Petri 网时,对于可触发的没有冲突的变迁,可直接将其设为 true,而对于有冲突的变迁,根据上文介绍的冲突竞争策略选择一个变迁改变状态为 true。

(2) 变迁的可触发时间赋值是根据用在时间区间上服从指数分布的函数生成的。

(3) 推进仿真时钟时,系统的仿真时钟等于上一次变迁的仿真时钟加上最短变迁触发时间,其他可触发变迁的时钟变量等于可触发时间减去最短变迁触发时间。

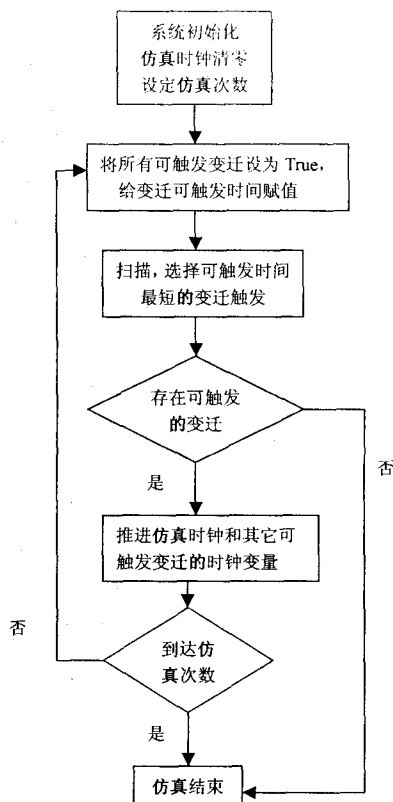


图 2 仿真流程图

4 仿真实例

下面以某计算机公司计算机销售过程的工作流为例来验证该流程的时间性能,实例的模型^[6]介绍:

(1) 用户通过 Internet 或其它方式向公司发出订单,指定所需要的计算机基本配置和数量,在随机 Petri 网模型中用变迁 t_0 表示;

(2) 公司接收用户订单,在模型中用变迁 t_1 表示;

(3) 公司对用户订单进行检查,即指以下 3 个活动的并发过程,可用一个瞬时变迁表示:

① 计算价格,在模型中用变迁 t_2 表示;

② 检查零部件库存是否满足需求,在模型中用变迁 t_3 表示;

③ 进行配置检查,确认用户的订单技术上是否可行,在模型中用变迁 t_4 表示。

(4) 进行决策,在模型中用变迁 t_5 表示:

① 如果通过检查,则继续执行;

② 如果未通过检查,则向用户发出信函,解释为什么订单不能完成,并提出修改意见,修改订单,在模型中用变迁 t_{10} 表示;接着交给办事员发出回馈意见,在模型中用变迁 t_{11} 表示;

(5) 准备接受订单的确认通知,即指一方面通知用户付款,另一方面发出生产通知的两个并行过程,它本身可用一个瞬时变迁表示,通知用户付款,在模型中用变迁 t_8 表示;

(6) 用户接到通知,进行付款,在模型中用变迁 t_9 表示;

(7) 发出生产通知,在模型中用变迁 t_6 表示;

(8) 装配计算机,在模型中用变迁 t_7 表示;

(9) 发货,在模型中用变迁 t_{12} 表示。

根据该 workflow 系统的随机 Petri 网模型如图 3 所示。

现假设变迁 $t_0, t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}$ 的平均延迟时间(单位时间为分钟)为 $1/\lambda_1 = 15, 1/\lambda_2 = 8, 1/\lambda_3 = 30, 1/\lambda_4 = 20, 1/\lambda_5 = 40, 1/\lambda_6 = 5, 1/\lambda_7 = 120, 1/\lambda_8 = 40, 1/\lambda_9 = 2880, 1/\lambda_{10} = 5, 1/\lambda_{11} = 40, 1/\lambda_{12} = 150$ 。在 t_5 处分之选择接受订单的概率 $\alpha = 0.91$ 。仿真运行 3 次,其中仿真执行了 10 000 次事件,平均延迟时间分别为 2937.8, 2989.5, 2928.7, 可以看到这与文献[6] 根据性能等价公式计算出来的总的性能等价时间 3043.3 十分接近。

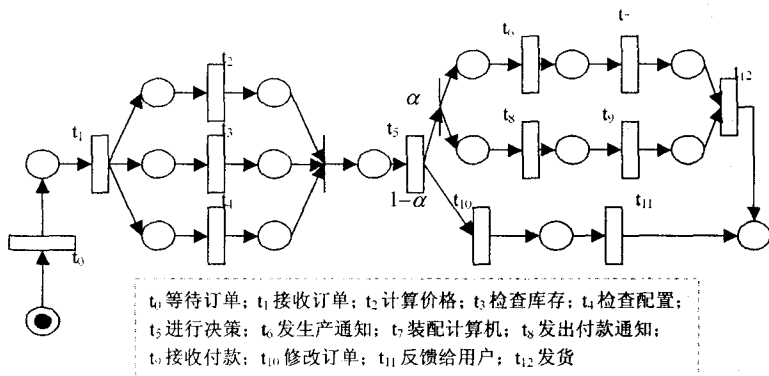


图 3 对应 workflow 模型的随机 Petri 网模型

5 结束语

workflow 模型的性能分析是 workflow 系统的一个非常重要的问题,但是对于一个实际的基于随机 Petri 网的工作流模型来说,状态空间往往是很大的,对其进行性能分析将非常困难。基于 Petri 网的工作流仿真技术简化了 workflow 仿真的集成。文中所设计实现的随机 Petri 网仿真工具为 workflow 系统分析提供了一种非常便捷的辅助手段。可进一步扩展这个仿真工具不仅适用于随机 Petri 网,也适用于各种时间 Petri 网。

参考文献:

[1] 袁崇义. Petri 网原理[M]. 北京:电子工业出版社,1988.

(下转第 94 页)

3 系统涉及的关键技术

3.1 多页面间共享数据

在 B/S 模式下, 用户发送请求及接受服务器端的请求都是基于 HTTP 协议的, 由于 HTTP 协议是无状态的, 因此请求信息和响应信息无法通过 HTTP 协议本身进行传递。为了跟踪用户的操作状态, 使用 Struts 中的 HttpSession 对象实现对会话状态的跟踪。具体实现代码如下:

HttpSession session = request.getSession();//创建与当前请求相关联的会话

```
DynaActionForm login = (DynaActionForm)form;
String userid = (String) login.get("userid");
String password = (String) login.get("password");
List userlist = module.login(userid, password);
if (userlist.size() > 0) {
    TbUser user = (TbUser) userlist.get(0);
    session.setAttribute("classid", user.getClassid());
    session.setAttribute("departid", user.getDepartid());
    .....
    return mapping.findForward("index");
} else {
    return mapping.findForward("login");
}
```

3.2 数据访问的实现

由于使用了 Spring + Hibernate 的机制, 这种机制主张接口编程, 充分利用 Java 中的继承机制, 基于这种情况笔者特地编写了一个操作数据库的类 BaseDao, 这样在后面的要操作数据库的类中, 只要用 Spring 的依赖注入机制就可以实现其它类操作数据库的功能。

BaseDao 类的主要方法如下:

```
public class BaseDao extends HibernateDaoSupport implements
IBaseDao {
    public void addObj(Object obj) {
        getHibernateTemplate().save(obj);
    }
    public void delObj(Class clazz, Integer ID) {
        Object obj = findById(clazz, ID);
        getHibernateTemplate().delete(obj);
    }
    .....
}
```

(上接第 90 页)

- [2] WfMC. Workflow management coalition terminology and glossary [EB/OL]. 1999. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.
- [3] VanderAast WMP, Hee K. 工作流管理—模型、方法和系统 [M]. 王建民, 闻立杰等译. 北京: 清华大学出版社, 2004.

4 结束语

通过实践证明, 整合框架可以提高系统代码的可重用性和控制层的灵活性, 大大提高了系统的稳定性、可扩展性和可维护性。文中的创新点在于: 把基于 MVC 模式的 Struts 框架和 Hibernate 框架灵活地应用到了一个在线作业管理系统, 并深入分析了系统涉及的关键技术部分 Struts-config.xml 配置文件及 Hibernate 框架的数据封装和 O/R 映射技术, 深入分析了该框架的核心配置文件 hibernate.cfg.xml。同时详细介绍了 Struts 框架的详细工作流程。后续研究目标是实现基于 Struts 和 Hibernate 整合框架的事务管理机制并提出一种合理的安全控制措施。

参考文献:

- [1] 李刚. Struts 2 权威指南: 基于 WebWork 核心的 MVC 开发 [M]. 北京: 电子工业出版社, 2007.
- [2] 计磊, 李里, 周伟. 精通 J2EE-Eclipse. Struts. Hibernate. Spring 整合应用案例 [M]. 北京: 人民邮电出版社, 2006.
- [3] 王嘉. Ajax 经典案例开发大全 [M]. 北京: 人民邮电出版社, 2007.
- [4] Malinescue F. EJB Design Patterns [M]. 北京: 机械工业出版社, 2004.
- [5] Allamaraju S, Buest C, Davies J. Professional Java Server Programming J2EE 1. 3 Edition [M]. 北京: 机械工业出版社, 2002.
- [6] 郭荷清, 林拉, 张为. 现代软件工程 [M]. 广州: 华南理工大学出版社, 2004.
- [7] Deitel H M, Deitel P J, Santry S E. Advanced Java 2 Plat2 form How to Program [M]. 北京: 电子工业出版社, 2004.
- [8] 随永, 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): 119-121.
- [9] 吴寅斐, 何钦铭, 吴大珊. Struts 分析及其应用研究 [J]. 计算机工程, 2006, 31(16): 42-44.
- [10] 吴昊, 李志蜀, 杨正海. 基于 Struts 框架构建 GSM 电子运行维护系统 [J]. 计算机应用与软件, 2005, 22(5): 63-65.
- [11] 杨涛, 周志波, 凌力. 基于 Struts 和 Hibernate 的 J2EE 快速开发框架的设计与实现 [J]. 计算机工程, 2006, 32(10): 82-85.

- [4] 王维平. 离散事件系统建模与仿真 [M]. 北京: 清华大学出版社, 1999.
- [5] 卢光松, 葛运建. 随机时间 Petri 网综述 [J]. 计算机科学, 2005, 32(3): 26-30.
- [6] 林闯, 田立勤, 魏丫丫. 工作流系统模型的性能等价分析 [J]. 软件学报, 2002, 13(8): 1472-1480.