基于工作流日志的决策规则挖掘研究 *

高 昂[1], 杨 扬[1], 张静乐[1], 王 薇[1,2]
(1. 北京科技大学 信息工程学院, 北京 100083; 2. 公安海警高等专科学校, 浙江 宁波 315801)

摘 要: 为了挖掘工作流日志中的决策规则信息,分析了工作流日志中的数据属性如何影响工作流实例的路径选择。基于算法挖掘工作流日志过程模型,对过程模型中的决策点进行分析,通过决策树分析技术结合工作流日志中的数据属性挖掘出影响工作流实例路由的决策规则。分析了现实应用中决策规则挖据所遇到的问题,并提出解决算法。最后通过测试程序测试并验证了挖掘过程。测试结果表明该算法能够正确地挖掘出决策规则。

关键词: 过程挖掘; 决策挖掘; 隐藏活动; 重复活动; 决策树; 决策规则
中图分类号: TP391 **文献标志码**: A **文章编号**: 1001-3695(2009)11-4104-04
doi:10.3969/j.issn.1001-3695.2009.11.031

Research on decision rule mining based on workflow logs

GAO Ang[1], YANG Yang[1], ZHANG Jing-le[1], WANG Yue-wei[1,2]
(1. *School of Information Engineering*, *University of Science & Technology Beijing*, *Beijing* 100083, *China*; 2. *Public Security Marine Police Academy*, *Ningbo Zhejiang* 315801, *China*)

Abstract: In order to mining the decision rule from workflow logs, this paper analyzed how data attributes influence the choices made in the process instances based on workflow logs. Mining process model in workflow logs based on α-algorithm, analyzed the decision point in process model, detection decision rule that affected the routing of a case through techniques based on decision tree analyze and data attributes in the workflow logs. Discussed problems related to real-life decision rule mining, presented an improved algorithm. To verify the presented ideas a test program had been implemented. The test result shows the alogrithm can correctly mining the decision rule from workflow logs.

Key words: process mining; decision mining; invisible activity; duplicate activity; decsion trees; decision rule

在 ERP、CRM、SCM 等信息处理系统[1]的运行过程中,会产生大量的工作流日志文件,这些日志文件中存着大量与过程相关的数据。过程挖掘[2,3]能够通过观察、分析工作流日志的数据和结构,自动化地构造过程模型。因为过程挖掘技术自动化构造出的过程模型能够反映出企业业务实际的执行过程,所以过程挖掘常常应用于企业工作流程的再造与诊断优化阶段,过程挖掘技术也大量应用于企业业务智能工具中。目前有很多过程挖掘算法专注于对工作流程控制流视图(control-flow perspective)的挖掘,如 α 算法[4]、α^{**} 算法[5]、启发式算法[6]等。很少关注工作流流程中各种数据属性的值对流程实例的路径选择所造成的影响。

工作流日志中包含了丰富的数据信息,如活动发生时间、活动执行者、活动的附加数据等。然而,目前大多数工作流挖掘技术仅仅利用了一到两种数据信息进行过程模型[4]构造。例如 α 算法通过分析日志中活动的发生时间进而得到活动间可能存在的依赖关系,生成活动的控制流视图,忽略了日志中与活动相关的其他数据信息。与过程模型关注控制流视图不同,决策规则挖掘关注的是工作流的案例视图(case perspective),分析日志中各种数据信息如何影响具体的过程实例(case)进行决策(例如在什么条件下执行活动 A,而不执行活动 B)。目前对决策规则挖掘方面的研究相对较少,文献[6]中提出了一种业务过程智能架构(business process intelligence, BPI),通过使用数据挖掘技术在一定范围内对业务过程行为进行分析。文献[7]使用决策树分析企业员工分配规则,并研究了从工作流日志中提取角色信息的方法。也有文献通过利用模糊事件和模糊函数将细胞神经网络算法整合到了基于事件驱动的过程链(event-driven process chain, EPC)模型中。本文论述了决策规则挖掘的总体流程。

1 决策规则挖掘总体流程

图 1 展示了决策规则挖掘的整体流程。首先,各种信息处理系统会产生大量的系统工作流日志,通过各种过程挖掘算法对系统工作流日志进行控制流模型挖掘(本文采用算法),最后结合控制流模型和系统工作流日志进行决策规则挖掘,挖掘出相应的决策规则。决策规则挖掘的目的是为了挖掘出规则用于解释在什么情况下活动 A 被选择执行,而不是选择执行活动 B。决策规则的挖掘流程如图 1 所示。

作为图 1 的描述,首先通过典型的过程挖掘算法(本例为 α 算法)对工作流日志进行挖掘,得到基于控制流视图的过程模型。其中工作流日志满足:a)每一个发生的事件都对应系统中定义好的一个活动(activity);b)每一个发生的事件都属于一个过程实例(case);c)事件是有序的,即使是并发的事件

收稿日期: 2009-03-07; **修回日期**: 2009-04-09 **基金项目**: 国家自然科学基金资助项目(60673160,60873192);国家"863"计划资助项目(2007AA01Z234)

作者简介:高昂(1981-),男,湖南长沙人,博士研究生,主要研究方向为工作流挖掘、电子商务、Web 服务(gaoang318@163.com);杨扬(1955-),男,教授,博导,主要研究方向为电子商务、图像处理、网格计算等;张静乐(1980-),男,博士研究生,主要研究方向为网格计算、电子商务等;王 薇(1980-),女,讲师,硕士,主要研究方向为电子商务、网格计算等.

也会被顺序地记录下来。图 2(a)为一个工作流日志的序列化表现形式,通过 case ID 将工作流日志分组为一个个的过程实例,除了活动间的顺序关系以外忽略了其他一切数据属性。基于以上的工作流日志通过 α 算法进行过程挖掘得到的过程模型如图 2(b)所示,此模型将作为输入用于决策规则挖掘阶段。

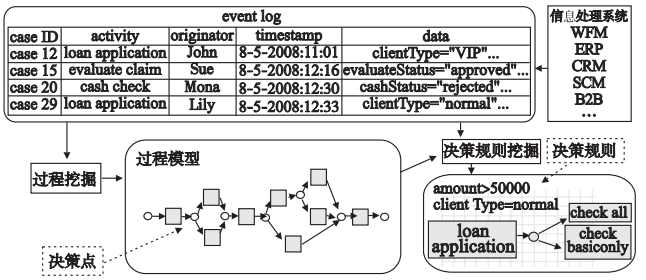
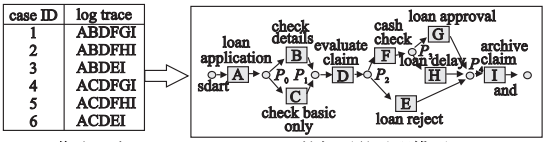


图 1　决策规则挖掘整体流程



(a) 工作流日志　　(b) 挖掘后的过程模型

图 2　过程挖掘阶段

这个过程模型主要反映了一个银行处理客户申请贷款的工作流处理过程:首先客户提交贷款申请 loan application(A),然后根据情况或者进行详细资料检查 check details 或者进行基本资料检查 check basic only(B 或 C)。之后进行风险评估 evaluate claim(D),接着或者执行拒绝放贷 loan reject 或者进行现金检查 cash check(E 或 F)。如果执行(F)之后就要选择或者执行贷款发放 loan approval 或者执行推迟贷款发放 loan delay(G 或 H)。最终工作流程以执行归档 archive claim(I)结束。

接着进入到决策规则挖掘的阶段,并将工作流日志中与活动相关的各种数据属性包括进来。图 3 为以 MXML[1] 格式展现的工作流日志截图。其中〈auditTrailEntry〉标签对应为一个活动,〈data〉标签对应为与活动相关联的附加数据信息。只有活动 A、D 和 F 与附加数据信息相关联。这里的附加数据信息是作为与活动相对应的数据属性的形式表示的,属性值可以写和修改。当执行活动 loan application 时,活动对应信息申请贷款的数额(amount),此次申请对应的申请号(applyID)和申请贷款客户的重要程度(clientType)被同时记录。当执行完活动 evaluate claim 时,评估结果被记录(evaluate_status),当执行完活动 cash check 时,现金检查结果被记录(cash_status)。从语义上分析,amount 是数值类型的属性,applyID 属性为惟一值用来惟一表示每一份贷款申请,clientType、evaluate_status 和 cash_status都是枚举类型属性(分别对应于 normal 或者 VIP, approved 或 rejected)。

可以看出,在过程挖掘阶段除了时间属性以外其他的工作流日志中的数据属性并没有被利用。同时通过过程模型也无法了解工作流日志中的数据属性如何影响流程实例的路径选择。就像图 1 所展示的那样,决策规则挖掘就是以基于控制流视图的过程模型和带有详细数据属性的工作流日志作为输入,经过一系列的处理,最终得到决策规则。

2　决策规则挖掘的主要步骤

为了分析商业过程中的选择流程,必须先对过程模型中的选择分支进行识别,也称为决策点识别;然后需要对与决策点相关的工作流日志中活动的数据属性进行分析得出决策规则。

图 3　MXML格式的工作流日志截图

2.1　识别过程模型中的决策点

定义 1　决策点。令 $N=(P,T,F)$ 为工作流网,$\exists p\in P$ 为决策点当且仅当 $|p\bullet|\geq 2$。

根据 Petri 网的定义[8],决策点对应于一个库所有多个输出弧。因为一个托肯只能被与输出弧所相连的某一个变迁所消耗,与之相对应的过程实例在此时某一条路径将被选择执行。在图 2(b)所展示的过程模型中存在三个决策点:P_0(如果 P_0 中存在一个托肯,B 和 C 只能有一个被选择执行)、P_2(如果 P_2 中存在一个托肯,E 和 F 只能有一个被选择执行)、P_3(如果 P_3 中存在一个托肯,G 和 H 只能有一个被选择执行)。

为了对商业过程中的流程选择作出分析,必须知道每一个过程实例在决策点所选择执行的分支。因此对应于工作流日志的可能的决策集必须被描述,决策能够通过对活动在过程模型的决策点处的执行情况进行观测。活动的执行情况映射为工作流日志中活动的发生轨迹[4]。一个过程实例的发生轨迹表明了这个过程实例在其对应的决策点处作出了决策。图 2(b)所展示的过程模型中所有的活动名称都对应于工作流日志中的事件名称。例如,一个活动实例的活动 loan delay 发生,那么 loan delay 将相应地记录在工作流日志中,同时表示这个活动实例在决策点 P_3 处作出了执行分支活动 loan delay 的决策。所以可以通过活动实例在每个决策点处所执行的第一个活动对可能发生的决策作出分类。

2.2　使用决策树分析过程模型的决策点

在对过程模型中的决策点进行完识别之后,需要判定过程实例的数据属性是否对实例的决策产生影响。例如过程实例具有某种特定的属性值将执行的特定的流程。机器学习技术[9]能够被用来挖掘结构化的模式基于分析训练实例集。

通过对决策点进行分析能够得到如图 4 所示的决策规则。在决策点 P_0 处,Check details(活动 B)被执行当且仅当 amount 大于 50 000 并且 clientType 为 normal,check basic only(活动 C)被执行当 amount 小于等于 50 000 或者 clientType 为 VIP。在决策点 P_2 处,cash check(活动 F)被执行当 evaluate_status 为 approved 时,loan reject(活动 E)被执行当 evaluate_status 为 rejected 时。在决策点 P_3 处,loan approval(活动 G)被执行当 cash_status 为 approved 时,loan delay(活动 H)被执行当 cash_status 为 rejected 时。

决策规则挖掘的总体思想就是将每一个决策点转换为一个分类问题[9~11],具体类别是能够作出不同的决策。本文使用工作流日志中的过程实例作为训练实例,对工作流日志中过程实例的数据属性的属性值进行分析。假设所有的过程实例的属性值在进行分支决策之前已被写入。对于决策点 P_0 只考虑活动 A 的数据属性(amount,applyID,clientType),每一个过程实例在工作流日志中所对应的属性值被用来创建训练实例(一个训练实例对应表中的一行),如图 5(a)所示。表的最后

一列代表(决策)类别,表示过程实例对应于决策点 P_0 所作出的决策(活动 B 和 C 中有一个被选择执行)。同样地,图 5(c)和(e)分别表示决策点 P_2 和 P_3 的训练实例。图 5(c)新添加了一个属性列(evaluate_status)由活动 D 所提供,因为活动 D 在决策点 P_2 之前执行。基于同样的考虑,图 5(e)新添加了属性列(cash_status)。类别列分别代表在决策点 P_2 和 P_3 处所作出的决策(分别为 E 或 F,G 或 H)。

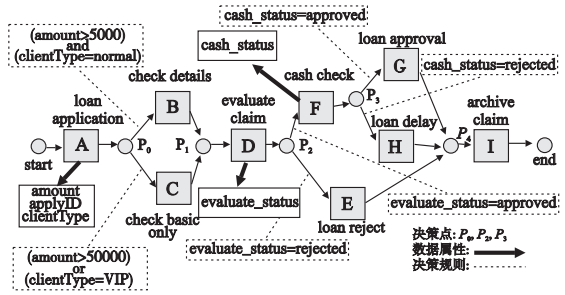
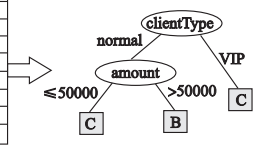


图 4　加入了决策规则的过程模型

applyID	amount	clientType	class
C567894938	80000	normal	B
C938609223	60000	normal	B
C135697567	55000	normal	B
C568120443	50000	normal	C
C493823084	20000	normal	C
C945675110	100000	VIP	C
C945675118	30000	VIP	C
C567897938	80000	normal	B

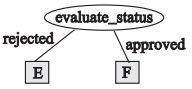
(a)决策点P_0的训练实例



(b)决策点P_0的决策树

applyID	amount	clientType	evaluate_status	class
C567894938	80000	normal	approved	F
C938609223	60000	normal	approved	F
C135697567	55000	normal	rejected	E
C568120443	50000	normal	approved	F
C493823084	20000	normal	approved	F
C945675110	100000	VIP	rejected	E
C945675118	30000	VIP	rejected	E
C567897938	80000	normal	approved	F

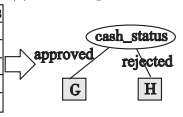
(c)决策点P_2的训练实例



(d)决策点P_2的决策树

applyID	amount	clientType	evaluate_status	cash_status	class
C567894938	80000	normal	approved	approved	G
C938609223	60000	normal	approved	rejected	H
C568120443	50000	normal	approved	approved	G
C493823084	20000	normal	approved	rejected	H
C567897938	80000	normal	approved	approved	G

(e)决策点P_3的训练实例



(f)决策点P_3的决策树

图 5　决策点的训练实例与其对应的决策树

有多种算法可以用来解决如上所述的分类问题[9,11]。本文使用决策树算法(C4.5[10])进行决策挖掘。决策树算法通过对根节点到叶节点的构造和排序对过程实例进行决策规则挖掘。图 5(b)(d)(f)显示了 P_0、P_2 和 P_3 所对应的决策树。

从决策树图 5(b)(d)(f)可用以下方法推导出图 4 所描述的决策规则。如果一个决策类别只出现在决策树的某一个叶节点上,那么这个类别满足从根节点到此叶节点的所有断言,断言之间用布尔表达式 and 连接。例如,在图 5(b)中类别 B 被选择如果(clientType = normal) and (amount > 50 000)。如果一个决策类别在决策树中被多个叶节点所表示的话,那么叶节点之间的断言通过布尔表达式 or 连接。例如,在图 5(b)中类别 C 被选择如果((clientType = normal) and (amount ≤ 50 000)) or (clientType = VIP),此表达式能化简为(clientType = VIP) or (amount ≤ 50 000)。

3　挖掘应用中的问题及解决算法

对实际的业务流程进行决策规则挖掘的过程中会面临两个主要问题:

a)工作流日志的数据质量好坏会对决策规则挖掘的结果造成直接的影响。现实中的工作流日志可能会产生数据丢失的情况,另外可能会记录不正确的数据,也就是产生所谓的噪声。挖掘算法必须具有足够的健壮性去处理带有噪声的工作流日志。另外,对数据属性的语义理解也存在偏差,如属性的真实意义、属性之间的相关程度等,目前还需要人工参与进行推理分析。目前存在一些半自动化的分析技术和辅助软件用来支持更加智能的决策规则挖掘。

b)对现实中的业务流程进行控制流程的挖掘建模时,会出现隐藏活动、重复活动问题。这些问题会对决策规则挖掘的正确性造成影响。

下面对一些重要概念进行定义。

定义 2　标签函数。令 T 为过程模型的活动集,L 为日志事件集,则 $l \in T \rightarrow L$ 为一个标签函数使每一个活动对应最多一个(1 或 0 个)日志事件。

以图 2(b)过程模型为例,所有的模型中的活动名都对应于其日志中的事件名。没有两个活动有相同的名称。另外,没有活动在工作流日志中没有记录。然而,在真实的过程模型中可能包含在日志中没有出现过的活动,如仅仅为了路由选择的目的而添加的活动。这些活动被称为隐藏活动。

定义 3　隐藏活动。一个活动 $t \in T$ 被称为隐藏活动当且仅当 $t \notin \mathrm{dom}\ (l)$。

图 6 展示了一个过程模型的片断,包含了一个决策点 P_1。P_1 的每个选择分支都是以一个隐藏活动开始的(黑色长方形表示)。因为这些活动不能从工作流日志中观察到,那么通过考察决策点的每一个选择分支的第一个发生的活动进而对可能的选择进行分类的方法在此处就失效了。

通过观察可知,活动 A 或 B 任意一个的发生说明第一个(最上面一个)选择分支被执行。同样地,活动 C 或 D 任意一个的发生说明第三个选择分支被执行。所以隐藏活动需要继续追踪其发生轨迹直至找到下一个非隐藏活动为止。这些非隐藏活动的发生说明了相应的选择分支被执行。

然而,追踪有时可能会进行得太远。图 6 中第二个选择分支的隐藏活动的后继活动为 F,但活动 F 的发生并不能说明第二个选择分支在 P_1 点被执行。因为在活动 F 之前是一个汇聚结构的库所,第一和第三个选择分支同样连接活动 F。同样地,活动 G 的发生不足以说明第四个选择分支被执行。因此,当遇到汇聚库所时应当停止追踪隐藏活动,丢弃相对应的选择分支。

另一个常见问题是过程模型中出现重复活动。这是由于在实际的过程模型中常常有多个活动对应于日志中同一个事件(不是一个单射函数)。

定义 4　重复活动。一个活动 $t \in T$ 被称为重复活动当且仅当 $\exists t' \in T$ 有 $t \neq t' \wedge l(t) = l(t')$。

图 7 展示了一个过程模型的片断,包含一个决策点 P_1,P_1 的每一个选择分支都以一个重复活动 A 开始,在第一个选择分支之后紧接着另一个重复活动 B。虽然重复活动(灰色表示)在工作流日志中有对应项,但是同样不能用来对决策点的可能的选择进行分类。

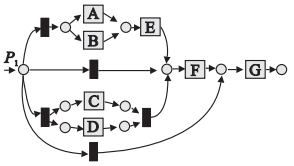


图 6　决策点附近有隐藏活动

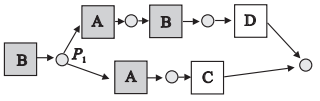


图 7　决策点附近有重复活动

一个可行的解决方案是用处理隐藏活动方法同样地处理重复活动,追踪重复活动的后继活动直到或者发现一个非重复非隐藏的活动或者发现一个汇聚结构的库所。在图 7 中,只有活动 C 和 D 能被用来确定一个过程实例在决策点 P_1 选择哪

一个分支执行。

算法 1　使用递归对过程模型的决策点的可能决策进行分类

```
miningDecisionClasses:
1. decisionClasses = 新的空集
2. while 决策点还有剩余的输出弧时 do
3.    currentClass = 新的空集
4.    t = 当前输出弧所对应变迁
5.    if (t≠隐藏活动) ∧ (t≠重复活动) then
6.       添加 l(t) 到 currentClass
7.    else currentClass = traceDecisionClass(t)
8.    end if
9.    if currentClass≠∅ then
10.      添加 currentClass 到 decisionClasses
11.   end if
12.   end while
13.   return decisionClasses
traceDecisionClass:
1. decisionClass = 新的空集
2. while 传递进来变迁的后继库所还有剩余时 do
3.    p = 当前的后继库所
4.    if p = = 汇聚结构类型 then
5.       return ∅
6.    else
7.       while p 的后继变迁还有剩余时 do
8.          t = 当前的后继变迁
9.          if (t≠隐藏活动) ∧ (t≠重复活动) then
10.            添加 l(t) 到 decisionClass
11.         else result = traceDecisionClass(t)
12.            if result = ∅ then
13.               return ∅ //(a)
14.         else result∪decisionClass
15.         end if
16.      end if
17.      end while
18.   end if
19. end while
20. return decisionClass //(b)
```

算法 1 描述了如何对基于 Petri 网过程模型的决策点进行决策集分类挖掘。算法的输入参数为具有多个输出弧的库所(决策点),然后将每一个输出弧视为一个选择分支(潜在的决策类),对每一个选择分支进行追踪,如果某个分支的第一变迁既不是隐藏活动也不是重复活动,那么与这个活动所对应的日志事件就能够被直接作为一个决策类。图 2(b)所示的过程模型有三个决策点,使用算法 1 进行挖掘后结果为:P_0 得到{{B},{C}},P_2 得到{{E},{F}},P_3 得到{{G},{H}}。

如果某个分支找到的第一个变迁是隐藏活动或者是重复活动,那么就必须继续追踪这个变迁的后继,如果变迁的后继为一个汇聚结构的库所,那么就将这个分支丢弃(a),否则继续追踪。如果变迁的后继为非隐藏非重复活动,那么就将这些活动归为一个决策类(b)。对图 6 的决策点 P_1 使用算法 1 得到的结果为{{A,B},{C,D}}。第二和第四个分支没有被算做决策类是因为在遇到任何非隐藏非重复活动之前就碰到了汇聚结构的库所,分支被丢弃。对图 7 的决策点 P_1 使用算法 1 得到的结果为{{C},{D}}。

4　程序实现

本文使用 Java 语言实现了对工作流日志进行决策规则挖掘的测试程序。挖掘过程为:以 MXML 标准格式的工作流日志作为输入,首先生成基于控制流视图的过程模型(算法),然后判定过程模型中的决策点,使用算法 1 对决策点可能的决策进行分类(解决了隐藏活动和重复活动的问题),结合工作流日志中相关的数据属性使用决策树算法(C4.5 算法)最终得出决策规则。图 8 和 9 是对第 2 章工作流日志进行决策规则挖掘的抓图,从图中可知程序正确地挖掘出了相应的决策规则。

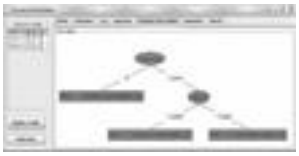


图 8　P_0点对应的决策树结果

图 9　最终生成的P_0点处的决策规则

5　结束语

本文详细介绍了对工作流日志进行决策规则挖掘的思想和方法。先对工作流日志进行控制流程视图的挖掘,生成过程模型,然后分析过程模型中的决策点,对可能的决策进行分类,结合工作流日志中相关的数据属性使用决策树算法最终得出决策规则。重点分析了决策规则挖掘在实际应用中所面临的问题,提出了对隐藏活动和重复活动进行挖掘处理的算法,并基于上述方法编写了决策规则挖掘的测试程序,程序正确地挖掘出了第 2 章所示的过程模型的决策规则。下一步的研究工作是对挖掘过程进行扩展,加入对更多过程模型的支持(如 EPC[11]),支持对噪声数据的处理,对更高层次的信息进行挖掘(如角色关系等[6])。

参考文献:

[1] DUMAS M, Van der AALST W M P, Ter HOFSTEDE A H M. Process-aware information systems: bridging people and software through process technology[M]. [S. l.]: Wiley, 2005.

[2] Van der AALST W M P, Van DONGEN B F, HERBST J, *et al*. Workflow mining: a survey of issues and approaches[J]. Data and Knowledge Engineering, 2003, 47(2):237-267.

[3] Van der AALST W M P. Business alignment: using process mining as a tool for delta analysis[C]// Proc of the 5th Workshop on Business Process Modeling, Development and Support (BPMDS'04). Latvia: Riga Technical University, 2004:138-145.

[4] Van der AALST W M P, WEIJTERS A J M M, MARUSTER L. Workflow mining: discovering process models from event logs[J]. IEEE Trans on Knowledge and Data Engineering, 2004, 16(9):1128-1142.

[5] 李嘉菲,刘大有,杨博. 过程挖掘中一种能发现重复任务的扩展 α 算法[J]. 计算机学报, 2007, 30(8):1436-1445.

[6] GRIGORI D, CASATI F, CASTELLANOS M, *et al*. Business process intelligence[J]. Computers in Industry, 2004, 53(3):321-343.

[7] LY L T, RINDERLE S, DADAM P, *et al*. Mining staff assignment rules from event-based data[C]//Proc of Business Process Management Workshops. Berlin: Springer-Verlag, 2005: 177-190.

[8] JENSEN K. Coloured Petri nets: basic concepts, analysis methods and practical use[M]. Berlin: Springer-Verlag, 1997.

[9] MITCHELL T M. Machine learning[M]. [S. l.]: McGraw-Hill, 1997.

[10] QUINLAN J R. C4.5: programs for machine learning[M]. [S. l.]: Morgan Kaufmann Publishers, 1993.

[11] WITTEN I H, FRANK E. Data mining: practical machine learning tools and techniques[M]. 2nd ed. [S. l.]: Morgan Kaufmann, 2005.

[12] ROZINAT A, Van der AALST W M P. Conformance testing: measuring the fit and appropriateness of event logs and process models [C]//Proc of Business Process Management 2005 Workshops. Berlin: Springer-Verlag, 2006: 163-176.