基于模拟退火算法的过程挖掘研究

宋 炜,刘 强
(清华大学软件学院软件工程与项目管理研究所,北京 100084)

摘 要: 模拟退火过程挖掘算法是为了更好地挖掘过程模型中非自由选择结构而提出的.模拟退火算法用于过程挖掘的基本思想,是以因果矩阵模拟事件日志行为,通过退火操作对因果矩阵进行处理,并对挖掘结果不断进行量化衡量.本文的主要工作包括:(1)在过程挖掘的环境下实现模拟退火算法;(2)用因果矩阵作为内部表示;(3)在退火操作选择过程中引入启发式规则;(4)对挖掘结果进行量化衡量,并通过过程挖掘的测试平台 Prom 进行实现和检测.

关键词: 模拟退火算法; 因果矩阵; 非自由选择结构; 过程挖掘
中图分类号: TP311.5 **文献标识码**: A **文章编号**: 0372-2112 (2009) 4A-135-05

Business Process Mining Based on Simulated Annealing

SONG Wei, LIU Qiang
(School of Software, Tsinghua University, Beijing 100084, China)

Abstract: To retrieve the non-free choice structure of the process model quickly and precisely, this paper propose a simulated annealing process mining approach to address this issue. Main contribution of the work includes: (1) Apply the simulated annealing approach under the setting of process mining. (2) Represent process model as "causal matrix". (3) Use heuristic rules to select annealing operations. (4) Evaluate the mining result with a quantitative measurement, incorporate the ideas above into existing simulated annealing algorithm to form an integrated solution. We give experimental results which produced by the ProM, a platform for business process mining.

Key words: simulated annealing; causal matrix; non-free choice structure; process mining

1 引言

为了更好地优化工作流程,降低运营成本,以便在竞争中获得更大的优势,越来越多的组织机构引入了信息感知系统,如企业资源计划(ERP)、工作流管理系统(WFM)等.然而,信息感知系统的建立是以充分了解企业的工作流程为前提的,对工作流程的获取主要面临以下困难:(1)由于企业各部门人员对工作流程有不同的看法,使得同一流程有不同的甚至相反的描述.(2)在机构复杂的企业中,获取全局的工作流程是极其费时费力的.因此过程挖掘就是为解决以上困难而提出的.

目前过程挖掘领域包括控制流挖掘、过程多方面挖掘、过程模型评估等多方面[1,2],其中控制流挖掘是最活跃的方向之一.控制流挖掘是指从日志中发现各活动的因果依赖关系,这一方向的很多算法都是为挖掘控制流中不同的结构而提出的[3,4].其中对于非自由选择结构的挖掘是控制流挖掘的重要任务之一,也是最难解决的任务之一.虽然目前很多算法已可以对非自由选择结构进行挖掘,如遗传算法[5,6]、α++算法[7]等.但是,这些算法在有些方面仍需改进.例如,遗传算法虽然能对隐含任务,非自由选择结构等复杂结构进行挖掘却非常耗时,而α++算法缺少抗噪音的能力,本文提出的基于模拟退火的过程挖掘算法旨在对此进行改进.

2 相关工作

过程挖掘的基本思想在 1995 年由美国新墨西哥州立大学的 Cook 教授提出[8],随后提出的大多数算法都是以解决控制流为目标的.德国乌尔姆大学的 Herbst 等人提出的方法具有处理重名任务的能力,作者同时开发了三个算法:MergeSeq、SplitSeq 和 SplitPar[9,10].以荷兰埃因霍温理工大学的 van der Aalst 所提出的α算法为基础,很多基于启发式规则的算法[11,12]在控制流挖掘方面也取得了很好的效果.文献[7]提出的α++算法在α算法的基础上制定了挖掘非自由选择结构的启发规则,在过程日志没有噪音的前提下,实现了对大多数非自由选择结构的挖掘.文献[5,6]讨论了如何将遗传算法应

收稿日期:208-10-22;修回日期:2008-12-26
基金项目:国家重点基础研究发展计划(973)(No.2004CB719400);国家自然科学基金(No.50519130);国家高技术研究发展计划(863)(No.2007AA01Z122,No.2007AA04Z135)

用到过程挖掘中,通过定义交叉、置换等过程遗传操作算子,最终演化出与日志非常吻合的过程模型. 遗传过程挖掘算法实现了用统一方法综合解决非自由选择结构、不可见任务和重名任务的挖掘问题.

随着过程挖掘研究的发展,出现了许多过程挖掘工具,其中 van der Aalst 教授等人开发了开源的过程挖掘框架 ProM[13],目前集成了超过 90 种挖掘、分析和转换插件,是目前过程挖掘领域比较完备的研究测试平台之一.

3 模拟退火挖掘算法

3.1 问题提出

非自由选择结构是控制流中一种常见的结构,具有非自由选择关系的活动在事件日志中不会以相邻的形式出现,因此也称该活动间具有间接依赖关系. 由于一个活动可能和日志中任何一个不相邻的活动间存在非自由选择关系,基于相邻关系的启发式规则很难发现此结构. 目前对这一结构的挖掘算法中效果较好的有α+ +算法和遗传算法.α+ +算法是专门针对非自由选择结构挖掘提出的,它针对非自由选择结构加入了更多的判定规则,然而对含有非自由选择结构的复杂过程模型挖掘效果并不理想,同时对其它复杂结构挖掘及噪音处理的能力不强;遗传算法作为最优化搜索算法,在很多情况下搜索空间过大,因此需要在运行效率上加以提高. 本算法以因果矩阵为基础,整合部分启发式规则,将模拟退火算法应用于过程挖掘中,从而实现对简单结构挖掘的同时,具有挖掘非自由选择结构及抗噪音的目的. 为了在运行效率上较遗传算法有所提高,本算法通过缩小初始搜索空间,加快收敛速度来降低运行时间.

3.2 算法基本思想

本算法首先根据输入的事件日志来构造一个因果矩阵,该因果矩阵是根据业务日志得到的数学模型,可以按照一定规则转换为 Petri 网[14]. 在初始化因果矩阵时,假设出现在日志中的所有活动间都存在特定的因果关系,由此定义出搜索空间. 然后通过不断的退火操作逐步改进或删除冗余关系,每次退火操作后,对日志进行重放,得出一个量化结果以此判断此次退火操作是否使得搜索向最优化方向收敛. 直到得出最终的因果矩阵,并对其进行优化得到与业务过程最为接近的模型. 算法具体过程如下所示:

为了引入较少的冗余关系,模拟退火整体过程共需进行两次. 第一次对基本结构和隐含任务及重名任务这些具有直接依赖性质的结构进行处理,第二次主要对非自由选择结构进行挖掘. 进行两次挖掘的好处有以下两方面:(1)第一次退火过程所要挖掘的结构通过日志中活动的相邻情况就可以得出,若同时对不相邻活动间添加依赖关系,则会影响这些结构的挖掘效果;同时,很多针对直接依赖关系的启发式规则也可以在此次退火中应用,以达到加快收敛速度的目的. (2)经过第一次退火过程,可以删除许多活动间的依赖关系,在此基础上,所添加的间接依赖关系也会相应减少,从而达到了缩小搜索空间的目的.

两次退火过程的区别在于因果矩阵的初始化方式不同. 第一次对因果矩阵进行初始化是在日志中出现的相邻的活动间建立因果关系. 由于第二次退火过程主要对非自由选择结构进行挖掘,即对间接依赖关系进行挖掘,所以第二次对因果矩阵初始化要在第一次挖掘结果的基础上,将活动与其所有后继活动(包括后继的后继活动)建立因果关系.

3.3 因果矩阵

本算法借鉴并改进了遗传算法的因果矩阵,本算法采用的因果矩阵为一个六元组 $CM=(A,C,I,O,M,L)$. 其中

* A 为活动的有限集

* $C\subset A\times A$ 是因果关系的有限集

* $I:A\rightarrow I(A)$ 为输入映射函数, $\forall(t)\in A, I(t)=\{p|\ p\in\cdot t\}$ 表示从活动 t 到其前驱活动集的映射

* $O:A\rightarrow O(A)$ 为输出映射函数, $\forall(t)\in A, O(t)=\{p|\ p\in t\cdot\}$ 表示从活动 t 到其后继活动集的映射

* $M=\{Parallel/Select\}$ 表示 $I(A)/O(A)$ 集合中子集间的关系. 若 $M=Parallel$,则 $I(A)/O(A)$ 中子集间关系为并行,但子集中元素间关系为选择;$M=Select$ 时,$I(A)/O(A)$ 中子集间关系为选择,子集中元素间关系为并行.

* $L:I(A)\times O(A)\ ?\ LS(A)$ 主要用来处理重名任务,将活动 A 前集和后集中的对应子集映射到 $LS(A)$ 中的特定标号上. 例如:$\{X\}\subset I(A)$,$\{B,C\}\subset I(A)$,$\{E,F\}\subset O(A)$,$\{G,H\}\subset O(A)$,在所有日志实例中,当且仅当 $\{X\}$ 和 $\{E,F\}$、$\{B,C\}$ 和 $\{G,H\}$ 分别同时出现,则 L 为 $\{(\{X\}\times\{E,F\})\rightarrow 1,(\{B,C\}\times\{G,H\})\rightarrow 2\}$

表 1 Petri 网对应的因果矩阵

TASK	I(TASK)	M(I)	O(TASK)	M(O)
A	{}	Select	{{B,E}{C,E}{F}}	Parallel
B	{{A}}	Select	{{D}}	Parallel
C	{{A}}	Select	{{D}}	Parallel
D	{{B,E}{C,E}{F}}	Parallel	{}	Parallel
E	{{A}}	Select	{{D}}	Parallel
F	{{A}}	Select	{{D}}	Parallel

元组 M 的提出,可以使因果矩阵具有更强的表现力,在与 Petri 网进行转化的过程中,会得到更简洁的结构. 这

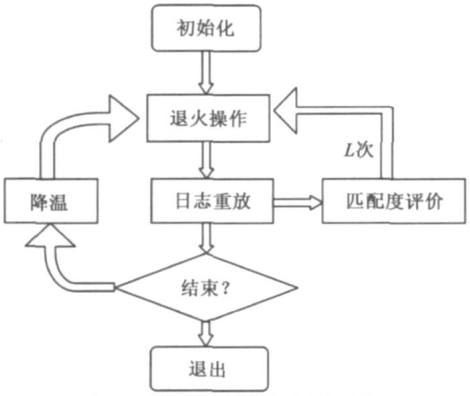


图1　模拟退火算法工作流程

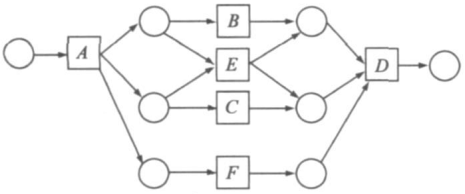


图2　理想化的工作流网

也是本算法中因果矩阵与遗传算法中因果矩阵主要区别.在元组L的定义上,本算法与遗传算法也有一定的差异.图2为原始日志对应的理想化 *Petri* 网,表1为按照上述规则得到的因果矩阵.

3.4　初始化

3.4.1　第一次退火过程的初始化

在初始化的过程中对所有日志进行遍历,根据所收集的活动相邻关系构造因果矩阵 $CM=(A,C,I,O,M,L)$,具体操作如下:

(1)将日志中的所出现的活动都加入到集合 A 中,构造有限活动集 A.

(2)若活动 a,b 在日志中相邻出现,则将 (a,b) 加入因果关系集 C 中.

(3)若 a,b 在日志活动中相邻出现(a 在前),则 $O(a)=O(a)\cup\{\{b\}\}$, $I(b)=I(b)\cup\{\{a\}\}$.通过不断更新活动 a 的 $I(a)$ 和 $O(a)$ 来构造映射 I 和 O.

(4)初始化时,设 $M(O(A))$ 为 Parallel(并行)标志, $M(I(A))$ 为 Select(选择)标志.

由 $M(I(A))$ 和 $M(O(A))$ 可知,初始化后 A 的前驱活动集 $I(A)$ 中的子集间均为选择关系,A 的后续活动集 $O(A)$ 中的子集间均为并行关系.根据 *Petri* 网理论可知,对于任意活动 $t\in A$, $I(t)$ 中的任何一个活动的执行都可以引发 t 的执行;t 的执行,可以使 $O(t)$ 中的所有活动处于可执行状态.因此初始化所得的模型所对应的 Petri 网在日志重放后会遗留大量使能标志.本算法引入的衡量标准就是以此为依据进行判断的.退火操作的最终目标就是在日志重放后,在因果矩阵所对应的 Petri 网中遗留的使能标志较上次达到最少.示例:令初始的事件日志 $LS=AFBCD,AFCBD,ABFCD,ACFBD,AEFD,AFED$.则按照上述规则初始化后得到的因果矩阵如下:

表2　初始化后的因果矩阵

	$I(TASK)$	$M(I)$	$O(TASK)$	$M(O)$
A	$\{\}$	Select	$\{\{B\}\{C\}\{E\}\{F\}\}$	Parallel
B	$\{\{A\}\{C\}\{F\}\}$	Select	$\{\{E\}\{D\}\{F\}\}$	Parallel
C	$\{\{A\}\{B\}\{F\}\}$	Select	$\{\{B\}\{D\}\{F\}\}$	Parallel
D	$\{\{B\}\{C\}\{F\}\}$	Select	$\{\}$	Parallel
E	$\{\{A\}\{F\}\}$	Select	$\{\{D\}\{F\}\}$	Parallel
F	$\{\{A\}\{B\}\{C\}\{E\}\}$	Select	$\{\{B\}\{C\}\{D\}\{E\}\}$	Parallel

3.4.2　第二次退火过程的初始化

在第二次退火过程的初始化时, $I(A)$ 和 $O(A)$ 的集合范围比第一次初始化大.因为在第一次初始化中,仅将 A 的直接前驱/后续活动加入到 $I(A)$ 或 $O(A)$ 中.第二次初始化要求将 $I(A)$ 中的所有活动的前驱集合并入到 $I(A)$ 中,将 $O(A)$ 中的所有活动的后继集合并入到 $O(A)$ 中.

设 $I'(A)$ 和 $O'(A)$ 为 A 的原前驱/后继活动集,用形式化语言描述为:

对任意活动 $B\in I'(A)$, $I(A)=I(B)\cup I'(A)$

对任意活动 $B\in O'(A)$, $O(A)=O(B)\cup O'(A)$

3.5　退火操作

退火操作应用于因果矩阵中的 $I(a)$ 和 $O(a)$,通过改变 $I(a)$ 和 $O(a)$ 中的元素以及子集间的关系使过程模型向最优化方向收敛.在每次退火时,以活动间关系在日志中出现次数为启发信息,以一定的概率选取如下操作之一:

(1)若 $(a,b)\in C$ 则 $C=C-(a,b)$,直接删除前后活动间的顺序关系.

(2)若 $(a,b)\in C$,且 $(a,c)\in C$,将 b 和 c 放入 $O(a)$ 的同一个子集,使 b,c 之间形成选择关系.

(3)若 $(b,d)\in C$,且 $(c,d)\in C$,将 b 和 c 放入 $I(d)$ 的同一个子集,使 b,c 之间形成并行关系.

设 $n(a,b)$ 为日志中活动 a 在活动 b 前出现的次数,启发信息如下:

(1)若 $|n(a,b)-n(b,a)|<\delta$,其中 δ 为一常量,且活动 a,b 存在共同后继活动 c,则以较大概率执行退火操作(3).

(2)若 $|n(a,b)-n(b,a)|>\delta$,且活动 a,b 存在共同前驱活动 c,则以较大概率执行退火操作(2).

(3)其它情况执行退火操作(1).

在退火过程结束后,若存在活动 t,使得对于任意 $I'(A)\subset I(A)$,都有 $t\in I'(A)$,则 $I(A)=\{t\}\cup\{I'(A)-\{t\}\}$, $M(I(A))=Parallel$;若存在活动 t,使得对于任意 $O'(A)\subset O(A)$,都有 $t\in O'(A)$,则 $O(A)=\{t\}\cup$

$\{O'(A)-\{t\}\}, M(O(A)) = Select$.

3.6 衡量标准

本算法中,衡量标准对退火的速率、算法的终止和挖掘结果的质量有很大的影响. 本算法通过对日志重放后的因果矩阵(Petri 网)所含使能标志的数量判断当前的退火操作能否被接受.

衡量标准分完整性和精确性两个方面. 完整性指过程模型可以匹配日志实例的程度;精确性指过程模型仅允许所给日志行为,而不能匹配日志以外的行为,即刻画实际工作流程的细致程度. 因此衡量标准为这两个方面的加权结合.

完整性标准 理想的完整性拟合需要过程模型匹配所有的日志实例. 过程模型能匹配的日志实例越多,其完整性拟合度越高. 为了能准确的说明完整性拟合度的定义,需要用到以下定义:

(1) allParsedTraces(L, CM) 表示可以被因果矩阵 CM 完全匹配的日志实例数.

(2) numTraces(L) 表示日志中所含的实例数.

(3) numActivitiesLeftTokens(σ_i, CM) 表示在重放日志实例 σ_i 时,所产生的使能标志未被消耗的活动数.

(4) numActivitiesInTrace(σ_i) 表示日志实例 σ_i 中所包含的活动数. 则完整性拟合公式为:

$$PF_{\text{complete}}(L, CM) = \frac{allParsedTraces(L, CM) - punishment}{numTraces(L)}$$

其中,

$$punishment = \sum_{i}^{numTraces(L)} \frac{numActivitiesLeftTokens(\sigma_i, CM)}{numActivitiesInTrace(\sigma_i)}$$

精确性标准 由于日志只提供了符合最终过程模型的日志实例,没有提供额外的不符合要求的实例(负例). 因此很难确定挖掘结果是否匹配日志以外的行为. 由于在过程模型中,过多的并行结构(顺序结构可看作一种特殊的并行结构)会导致模型过于独特,不具有普遍性,而选择结构过多会使得模型对日志刻画程度不够,则精确性标准可定义如下:

(1) AndRelations(L, CM) 表示在退火的某个阶段中,过程模型所含的并行结构数.

(2) OrRelations(L, CM) 为退火的某个阶段中,过程模型所含的选择结构数. 则精确性拟合度为:

$$PF_{\text{precise}} = \left| \frac{AndRelations(L, CM)}{AndRelations(L, CM) + OrRelation(L, CM)} - P \right|$$

其中 p 为算法输入参数,表示过程模型中包含并行结构的先验概率.

衡量标准 最终衡量标准为完整性标准和精确性标准的加权结合. 令 L 为一非空的事件日志, CM 为一因果矩阵. ψ为一个 0 到 1 的实数. 则最后的衡量标准 $F(L, CM)$ 的定义为:

$$F(L, CM) = PF_{\text{complete}} \cdot \psi PF_{\text{precise}}$$

4 实验结果分析和验证

4.1 时间复杂度分析

设 m 为日志 L 中所含日志实例数, n 为最长的日志实例所包含的任务数, p 为日志 L 中所出现的活动总数. q 为退火操作迭代次数. 本算法在第一次模拟退火过程中初始化时间复杂度为 $O(mn)$,退火过程时间复杂度为 $O(p^2q)$,结果转化时间复杂度为 $O(p^2q)$,日志重放时间复杂度为 $O(mn)$,所以第一次退火总时间复杂度为 $O(mn + p^2q)$. 在第二次退火过程的初始化阶段的时间复杂度与第一次不同为 $O(mnp)$. 所以,第二次退火过程的时间复杂度为 $O(mnp + p^2q)$. 算法总时间复杂度为 $O(mnp + p^2q)$.

4.2 实验结果比较

本算法在 ProM 平台进行了初步的实现和验证,所用实验数据部分来自 http://prom.win.tue.nl/tools/prom/ 上 ProM 平台所用测试数据,部分由 ProM Import Framework 创造. 最终结果不仅包含了基本结构和非自由选择结构,还包含了隐含任务和不可见任务. 由于模拟退火过程具有随机性和其空间最优化搜索的本质,本算法较α++算法具有抗噪音能力.

在运行效率方面,本算法只对一个因果矩阵进行操作和衡量,从而省去了在遗传算法中对多个因果矩阵进行操作、衡量和比较所用的时间,因此在对很多日志的挖掘中更有效率. 图 3 是两种算法执行时间的比较,从图中可以看到,在处理相同的事件日志时,遗传算法需要更多的执行时间. 因此模拟退火算法的执行效率要优于遗传算法.

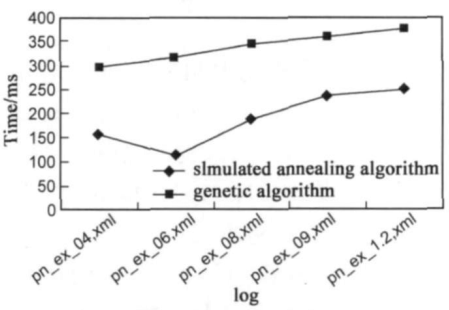


图3 模拟退火算法和遗传算法执行时间比较

对于部分复杂日志,本算法的挖掘结果较遗传算法与α++算法差,特别是对多种复杂结构并存的过程模型挖掘中,本算法还需进一步完善.

5 结论和展望

基于模拟退火的过程挖掘算法不但可以对并行、选择等基本的工作流结构进行挖掘,也可以对重名任务、隐含任务和非自由选择任务进行挖掘. 在对非自由

选择结构挖掘方面,由于其搜索空间小于遗传算法的搜索空间,因此,在执行效率上要高于遗传算法.但对很多复杂的工作流程所产生的日志,本算法不能取得理想的效果.本算法时间复杂度随日志中所含活动数目呈指数增长,在日志所含活动数较多的情况下,算法运行所用时间仍然很长.

对于同一个日志行为,可用不同的工作流结构进行表示,目前模拟退火过程挖掘算法还无法在拟合日志行为的基础上,对过程模型结构进行进一步优化.以上这些问题都需在以后的工作中加以解决.

参考文献:

[1] W M P van der Aalst, H A Reijers, A J M M Werjters, B F van Dongen, A K Alves de Medeiros, M Song and H M W V erbeek. Business process mining: an industrial application[J]. Information System, 2007, 32(5): 713 - 732.

[2] W M P van der Aalst, Minseok Song. Mining social networks: uncovering interaction patterns in business processes[A]. Business Process Management: 2nd International Conference, BPM 2004, Potsdam, Germany [C]. Berlin: Springer, 2004. 244 - 260.

[3] W M P van der Aalst, A J M M Weijters, L Maruster. Workflow mining: Discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128 - 1142.

[4] W van der Aalst. The application of petri nets to workflow management[J]. The Journal of Circuits, Systems and Computers, 1998, 8(1): 21 - 26.

[5] A K Alves de Medeiros, A J M M Weijters, W M P van der Aalst. Genetic process mining: an experimental evaluation[J]. Data & Knowledge Engineering, 2007, 14(4): 245 - 304.

[6] W M P van der Aalst, A K Alves de Medeiros, and A J M M Weijters. Genetic Process Mining[A]. Applications and Theory of Petri Nets: 26th International Conference ICATPN 2005, Miam USA[C]. Berlin: Springer, 2005. 48 - 69.

[7] Lijie Wen, Wil M P van der Aalst, Jianmin Wang, Jiaguang Sun. Mining process models with non-free-choice constructs [J]. Data Min Knowl Disc, 2007, 15(2): 145 - 180.

[8] J E Cook, A L Wolf. Automating process discovery through event-data analysis[A]. Proceedings of the 17th international conference on Software engineering[C]. Washington, USA: Association for Computer Machinery, 1995. 73 - 82.

[9] M Hammori, J Herbst, N Kleiner. Interactive workflow mining [A]. Proceedings of the 2nd International Conference on Business Process Management[C]. Berlin: Springer, 2004. 211 - 226.

[10] M Hammori, J Herbst, N Kleiner. Interactive workflow mining-requirements, concepts and implementations[J]. Data and Knowledge Engineering, 2006, 56: 41 - 63.

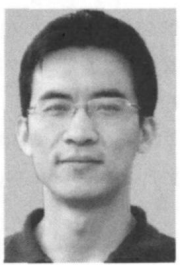
[11] Joonsoo Bae, LingLiu, James Caverlee, William B Rouse. Process mining, discovery, and integration using distance measures [R]. Atlanta, USA: Georgia Institute of Technology, 2006.

[12] Laura Maruster, A J M M Ton Weijters, W M P van der Aalst and Antal van den Bosch. Process mining: discovering direct successors in process logs[A]. Discovery Science: 5th International Conference, DS 2002, Lübeck, Germany [C]. Berlin, Springer, 2002. 364 - 373.

[13] B F van Dongen, A K Alves de Medeiros, H M W Verbeek, et al. The ProM framework: a new era in process mining tool support[A]. Application and Theory of Petri Nets[C]. Berlin: Springer-Verlag, 2005. 444 - 454.

[14] T Murata. Petri nets: Properties, analysis and applications[J]. Proceedings of the IEEE, 1989, 77(4): 541 - 577.

作者简介:



宋　炜　男,1983 年出生于内蒙古赤峰市.2004 年毕业于烟台大学计算机学院,其后在天津辰鑫石化设计有限公司工作,2006 年进入清华大学软件学院攻读硕士研究生.主要研究领域为过程挖掘、工作流技术.

E-mail: w-song06@mails.tsinghua.edu.cn

刘　强　女,1963 年出生于内蒙古包头市.清华大学软件学院副教授,主要研究领域为协同工作、工作流技术、需求工程.