基于初始不可见任务的过程挖掘方法的改进

薛 岗[1] 刘芳妤[1] 周逸璇[2] 叶小虎[2]
(云南大学国家示范性软件学院 昆明 650091)[1] (云南大学信息科学与工程学院 昆明 650091)[2]

摘 要 过程挖掘对于部署新的商业流程以及审计、分析和改进已有的流程是非常有帮助的。将解决在处理不可见任务时都存在的一些问题。例如一个任务存在于过程模型中,但是并没有在事件日志中体现,这样的任务主要服务于路由,但是绝对不能忽视。分析日志中不可见的任务,并对任务的强循环、弱循环等特性进行分析,在此基础上提出新的过程挖掘方法,以便更好地提高对不可见任务的挖掘效率。

关键词 任务,不可见,跳,循环,挖掘

Towards Improving Process Mining Results Based on Invisible Tasks

XUE Gang[1] LIU Fang-yu[1] ZHOU Yi-xuan[2] YE Xiao-hu[2]
(National Pilot School of Software, Yunnan University, Kunming 650091, China)[1]
(School of Information Science and Engineering, Yunnan University, Kunming 650091, China)[2]

Abstract Process mining is helpful for deploying new business processes as well as auditing, analyzing and improving the already enacted ones. The following text will solve the problem in dealing with invisible tasks, such tasks that exist in a process model but not in its event log. This is a problem since invisible tasks are mainly used for routing purpose but must not be ignored. Analyzed the features of strong-loop, weak-loop, jump and invisible tasks, on the basis of which a new algorithm was proposed. This algorithm can improve the efficiency of the mining of invisible tasks.

Keywords Task, Invisible, Jump, Loop, Mining

1 介绍

过程挖掘的目的在于从现有的事件日志中自动发现有价值的信息。第一个专门用于过程挖掘的算法在文献[1,2]中被提出。这些被挖掘出的信息可以用作部署新的系统去支持商业流程的执行,或者作为一种设计的分析工具。目前国内外很多学者致力于流程挖掘的研究,很多人采用WF网表示工作流模型,在此基础上提出了α算法、β算法。α算法的局限性在于不能挖掘循环、不可见任务、非自由选择等复杂结构。α+算法和α++算法在此基础上做了扩展,解决了部分问题,但是没有考虑活动执行的开始事件和结束事件之间的时间间隔问题。β算法考虑了时间间隔问题,但是不能够挖掘循环和不可见任务。

不可见任务是指那些存在于过程模型但是不存在于事件日志中的任务,这些任务是非常难挖掘的,因为他们不存在于事件日志的任何事件序列中,但是他们在现实的流程模型中是普遍存在的。国外学者W. M. P. van der Aalst已经做了大量关于不可见任务挖掘的工作,但是仍还存在很多问题没能得到解决,许多方法的挖掘效率都不是很理想。这篇文章将改进提高对不可见任务的挖掘效率。

2 相关工作

在文献[8]中介绍,流程挖掘算法的核心就是利用流程日志构建流程模型,Afrawal提出了基于活动间依赖关系的模型语言,该算法利用工作流管理系统的日志挖掘工作流模型。Herbest提出了一种基于ADONIS模型的算法,这个算法能够挖掘平行结构和选择结构。运行这个算法时,相同的任务会被赋予不同的名称,因此可以分析结构中包含相同名称活动的流程,但是它不能够发现隐藏结构,而且挖掘循环结构的能力也非常有限。

学者W. M. P. van der Aalst已经做了大量关于不可见任务挖掘的工作,他提出了一种名叫$\alpha^{\#}$的算法,能在DIWF-nets中很好地挖掘不可见任务。与遗传算法相比较,该算法能很好地发现不可见任务,但是仍然存在一些问题,例如挖掘的效率不是很高等。这篇文章将提高挖掘的效率。

ProM[9]是一个很好用的流程挖掘软件,它可以根据日志文件使用不同的算法挖掘流程模型,分析日志中不同任务的各种状态属性等,例如完成、不完成、事件的使用频率等,每个任务的数量也可以分析出来。这里将使用它作为实验工具。

3 基础技术

遗传挖掘算法是唯一原本就支持对不可见任务进行检测的方法,它运用了遗传算法的基本方法并且定义了过程挖掘的两种方法,旨在支持当前商业过程的所有控制流结构,尤其是循环任务、不可见任务、非确定平行选择结构。但是这个算

本文受云南省教育厅科学研究基础基金项目(2010Y252)资助。

薛 岗(1977-),男,博士,主要研究方向为工作流的编制与管理,E-mail:xiaohuye@live.cn;刘芳妤(1987-),女,硕士,主要研究方向为网络分布式、工作流;周逸璇(1987-),女,硕士,主要研究方向为流程挖掘;叶小虎(1987-),男,硕士,主要研究方向为流程挖掘。

法需要大量用户自定义的参数，而且不能总是确保在有限的时间内返回最恰当的结果。

不可见任务是指在任何日志轨迹中都不会出现的任务。导致不可见任务存在的情况有两种：1)任务在日志中不注册，不属于业务流程中的实际活动。因此在构造工作流网的时候常带来不能正确挖掘不可见任务的缺陷。WF_net 中存在两种不可见的任务：连接的两个不同的库所之间不存在其他的任务，如图 1 所示；连接的两个库所间存在其他的任务，如图 2 所示。2)日志中有噪声，真实的任务在日志轨迹中丢失。下面将详细分析日志中的复杂结构，使得在事件日志中不能体现的不可见任务能够很好地被挖掘出来。

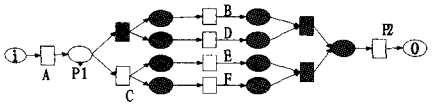

图 1　连接两个库所的任务

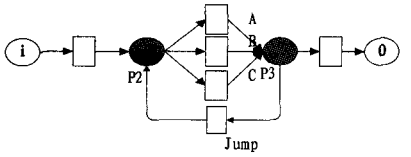

图 2　连接两个不通的库所之间存在的任务

日志中反复出现的任务为一步循环。如图 3 所示，完整的工作流日志是{AC，ABBBC}，B 是可以重复发生的，也是可以不发生的，循环任务可以在日志中重复出现也可以不出现。下面给出循环的定义。

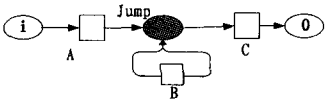

图 3　循环结构

定义 1　设 $N=\{P,T,V:F,K,R,OP\}$ 是一个同步管理工作流网，N 是循环结构的同步工作流网，当且仅当 $\forall t\in T:t\cdot\bigcap\cdot t\neq\Phi$。

在实际流程中，有些任务可能会向前或后跳到某个点重复执行任务，文献[4]证明了流程中“跳”结构的存在，虚变迁类似 WF_net 中的第二类不可见任务，即发生了“跳”的两个库所存在其他任务。因此挖掘不可见就是挖掘流程中的跳结构。

首先我们定义日志文件中事件之间的偏序关系。

1)$a\rightarrow_w b$，当且仅当 $a>_w b$ 且 $\wedge\neg(b>_w a)$ 或者 $a\infty_w b$。

2)$a\|\|_w b$，当且仅当 $a\times_w b$，且 $a\triangledown_w b$，或者 $a\triangle_w b$。

3)$a\rightarrow\!\!\!\!\cdot\ _w b$，当且仅当 $\forall\sigma\in W,\forall_{a,b}\in_T(a\in\sigma\wedge b\notin\sigma)\vee(b\in\sigma\wedge a\notin\sigma)$，表示 ab 互斥。

4)$a\triangleleft_w b$，当且仅当 $a\rightarrow\!\!\!\!\cdot\ _w b$，$a\triangledown_w b$。$a\triangleright_w b$ 当且仅当 $a\rightarrow\!\!\!\!\cdot\ _w b$，$a\triangle_w b$。

5)$a\|_{>w}b$ 当且仅当使得 $\neg(a\times_w b)$，$a>_w b$，$b>_w a$，且存在 $\exists c,d\in\sigma,c\ \#_w b$ 使得 $a\triangledown_w b$，$a\triangle_w b$。

6)$a\triangleleft\|\triangleright_w b$ 当且仅当 $\forall\sigma_i\notin W$，使得 $a\|_w b$，$a\|_{>w}b$，$a\triangleleft_w b$，$a\triangleright_w b$。

7)$a|\rightarrow_w b$，当且仅当 $\exists\sigma\in W_a,b\in\sigma$ 使得 $\forall t,t'\in T\ a\rightarrow_w t\wedge t'\rightarrow_w b\wedge\neg(t'\rightarrow_w t)\wedge(t)\gg_w t'\vee t\rightarrow_w t')$，$a\rightarrow_w b$。

8)$a|\leftarrow_w b$ 当且仅当 $\forall\sigma\in W,a,b\in\sigma$ 使得 $a\gg_w b$ 以及 $b\rightarrow_w a$ 且对于 $\forall_{c,d}\in\sigma c\rightarrow_w a\wedge\neg(c\rightarrow_w d)\ \wedge b\rightarrow_w d$。

9)$a\infty\infty$，当且仅当 $\sigma=e_1,e_2,\cdots,e_i,\cdots,e_n,i\in1,n-2,\sigma\in W$，如果存在 $e_i,task=e_{i+1},task=\cdots=e_{i+2x+1},task=a,x\in(1,\cdots,n)$，则单步弱循环 x 次。

10)$a\infty\infty$，当且仅当 $\forall\sigma\in W$，一定有 $e_i,task=e_{i+1},task=e_{i+2},task=e_{i+3},task=a$ 且 $\forall_{c,d}\in\sigma c\rightarrow_w a\wedge(\neg(c\rightarrow_w d))$ 表示单步强循环。

11)$a\underline{\infty}_w b$ 当且仅当 $a\infty_w b$ 且 $\forall\sigma\in W,a,b\in\sigma$ 使得 $\forall_{c,d}\in\sigma c\rightarrow_w a\wedge(\neg(c\rightarrow_w d))$；否则为弱循环，表示为 $a\infty_w b$。

$a\rightarrow_w b$ 表示任务间的直接后继关系。$a\triangledown_w b$ 表示任务间的分叉关系。$a\triangle_w b$ 表示任务间的聚合关系。$a\|_w b$ 表示任务间的平行关系。$a\ \#_w b$ 表示任务无关。$a\neg_w b$ 表示任务互斥关系。$a\triangleleft_w b$ 表示任务的选择分叉关系。$a\triangleright_w b$ 表示任务的选择聚合关系。$a\|_{>w}b$ 表示任务的交替平行关系。$a\triangleleft\|\triangleright_w b$ 表示任务的非确定平行关系。$a|\rightarrow_w b$ 表示后跳关系。$a|\leftarrow_w b$ 表示前跳关系。

4　挖掘算法

归纳事件日志中任务发生的特征，总结跳结构的判定定理如下。

定义 2　设 $N=\{P,T,V:F,K,R,OP\}$ 是一个同步管理工作流网，W 是 N 的一个完备的工作流日志，对于任意的 $a,b\in T:a|\rightarrow_w b$ 表示存在 $tjump\in Tjump$，使得 $p=a\cdot=\cdot tjump,p\cdot op=p\cdot op\cup jump$ 且 $p'=\cdot b=tjump=p'\cdot op=p\cdot op\cup xjoin$。

定义 3　设 $N=\{P,T,V:F,K,R,OP\}$ 是一个同步管理工作流网，W 是 N 的一个完备的工作流日志，对于任意的 $a,b\in T:a|\leftarrow_w b$ 表示存在 $tjump\in Tjump$，使得 $p=\cdot a=tjump\cdot,p\cdot op=p\cdot op\cup xjoin$ 且 $p'=b\cdot=\cdot tjump,p'\cdot op=p'\cdot op\cup xjoin$。

定义 4　设 $N=\{P,T,V:F,K,R,OP\}$ 是一个同步管理工作流网，W 是 N 的一个完备的工作流日志，对于任意的 $a,b\in T:a\infty1_w$ 表示存在 $tjump\in Tjump$，使得 $p=\cdot a=tjump\cdot,p\cdot op=p\cdot op\cup xjoin$ 且 $p'=\cdot a=\cdot tjump,p'\cdot op=p'\cdot op\cup xjoin$。

定义 5　设 $N=\{P,T,V:F,K,R,OP\}$ 是一个同步管理工作流网，W 是 N 的一个完备的工作流日志，对于任意的两个任务集 PS,SS 使得 $PS\subseteq T,SS\subseteq T$。

1)若 $\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b\wedge a|\rightarrow_w b$，iff $\exists_{p,p'}\in_P\forall_a\in_{ps}\forall_b\in_s p=a\cdot=\cdot tjump\wedge p'=\cdot b=tjump\cdot,p\cdot op=p\cdot op\cup xjoin$。

2)若 $\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b\wedge b|\rightarrow_w a$，iff $\exists_{p,p'}\in_P\forall_a\in_{ps}\forall_b\in_s p=a\cdot=\cdot tjump,p\cdot op=p\cdot op\cup xjoin$，且 $p'=\cdot a\cdot=\cdot tjump,p'\cdot op=p'\cdot op\cup jump$。

3)若 $\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b\wedge b\underline{\infty}_w a$，iff $\exists_{p,p'}\in_P\forall_a\in_{ps}\forall_b\in_s p=a\cdot=\cdot tjump,p\cdot op=p\cdot op\cup xjoin$，且 $p'=\cdot a\cdot=\cdot tjump,p'\cdot op=p'\cdot op\cup jump$。

4)若 $\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b,\forall_{a1,a2}\in_{PS}a_1\|_w a_2$，iff $\exists_{p,}\in_P\forall_a\in_{ps}\forall_b\in_s a\cdot\bigcap\cdot b=\{p\},p\cdot op=\{p\cdot op\cup join\};\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b,\forall_{b1,b2}\in_{PS}b_1\|_w b_2$，iff $\exists_p\in_P\forall_a\in_{ps}\forall_b\in_s a\cdot\bigcap\cdot b=\{p\},p\cdot op=\{p\cdot op\cup\text{and}\}$。

5)若 $\forall_a\in_{ps}\forall_b\in_s a\rightarrow_w b,\forall_{a1,a2}\in_{PS}a1\triangleright_W a_2$，iff $\exists_p\in_P$

$\forall_a \in_{ps} \forall_b \in_{ss} a \cdot \cap \cdot b$,若$|a|>1, p \cdot op=\{xjoin \cup p \cdot op\}$；若$\forall_a \in_{ps} \forall_b \in_{ss} a \rightarrow_w b$, $\forall_{b1,b2} \in_{PS} b1 \lhd_w b_2$, iff $\exists_p \in_P \forall_a \in_{ps} \forall_b \in_{ss} a \cdot \cap \cdot b$,若$| b | > 1\ p \cdot op=\{p \cdot op \cup or\}$。

上面仔细分析了模型中可能存在的各种结构，基于定义1—定义5给出如下的挖掘算法。

1) $T_W=\{t \in \sigma | \sigma \in W\}$

2) $T_I=\{first(\sigma) | \sigma \in W\}$

3) $T_O=\{lost(\sigma) | \sigma \in W\}$ |

4) $LIL=\{ t \in T_{log} | \exists \sigma=e_1, e_2, \cdots, e_i, \cdots, e_n \in W[e_i \cdot task = e_j \cdot task = e_k \cdot task = t \wedge l=k+1=j+2=j+3]\}$

5) $T'=T_W | L1L, L2L=\Phi; F_{L1L}=0$

6) For each $t \in L1L$ do

1. $A=\{a \in T' | a >_w t\}$;

2. $B=\{ b \in T' | t >_w b \}$;

3. 如果$\forall a_1 \in A, b_1 \in B\ a_1 >_w b_1 \Rightarrow F_{L1L}:=F_{L1L} \cup \{(t, p(A|B, B|(L1L:=L1L-1)) \wedge (L2L=L2L \cup t) \wedge F_{L1L}:=F_{L1L} \cup \{(tjump, p(A, t)), (p(t, B), tkump)\}$

7) $W^{-L1L}=\Phi$

8) For each $\sigma \in W$ do

1. $\sigma'=\sigma$;

2. For each $t \in$ L1L do $\sigma'=$remove Task(σ', t), $T_W=T_W - t$;

3. For each $t \in$ L2L do $\sigma'=$remove Task$(\sigma', t, 1)$;

4. $W^{-L1L}=W^{-L1L} \cup \sigma'$。

9) $W=W^{-L1L}$

10) $X_W=\{\langle ps, ss, op\rangle | ps \subseteq T_w \wedge ss \subseteq T_w \wedge op \in op \wedge \forall_a \in_{ps} \forall_b \in_{ss} a \rightarrow_w b \wedge op=$default$\}$

11) $D_S=\{(t_{(Pin, Pout)}; t_{(P'in, P'out)}) | (Pin; Pout); (P'in; P'out) \in Y_I \cup Y' \wedge Pout \cap P'in \neq \Phi\} \cup \{(a, t_{(Pin, Pout)}) | (Pin; Pout) \in Y_I \cup Y' \wedge \exists (A; X) \in P_{in}: a \in A\} \cup \{(t_{(Pin, Pout)}; b) | (P_{in}; P_{out}) \in Y_I \cup Y' \wedge \exists (Y; B) \in P_{out}: b \in B\}$

12) $D_P=\{(t_{(Pin, Pout)}; t_{(P'in, P'out)}) | (Pin; Pout); (P'in; P'out) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; (A'; X') \in P'_{in}: \exists a \in A; a' \in A'; x \in X; x' \in X': a \| W a' \vee x \| W x'\} \cup \{(t; t_{(Pin, Pout)}) | t \in T_W \wedge (Pin; Pout) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; \exists a \in A; x \in X; a \| W t \vee x \| W t\} \cup \{(t; t_{(Pin, Pout)}) | t \in T_W \wedge (Pin; Pout) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; \exists a \in A; x \in X; a \| W t \vee x \| W t\}$

13. $T_W=T_W \cup \{t_{(Pin, Pout)} | (Pin; Pout) \in Y_I \cup Y_1'\}$

14. $(W)=(T_W; T_I; T_O; D_S; D_P)$。

第1,2,3步得到事件日志中的所有任务，发现初始任务及结束任务。第4步找到一步循环任务。第5,6步构建一步弱循环和强循环之间的流关系。第7,8,9步去掉轨迹日志中的弱循环任务，保持一个强循环的任务重新构建轨迹日志。第10步从日志中获取直接的后继关系集合。第11,12步添加新的因果联系。第13,14步扩建任务序列的不可见任务。

5 实验分析

这里使用ProM作为日志挖掘的工具，使用电话公司的维修日志作为实验日志，运行新的挖掘算法得到过程模型，如图4所示。

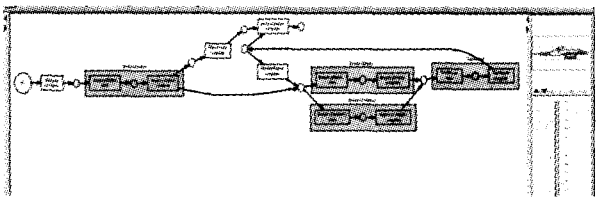

图4 流程模型

算法主要改进对日志轨迹中不可见任务的挖掘，为了比较和遗传算法的差异，我们在20组人造日志上分别进行实验，得到图5的比较结果，清晰地表示出对不可见任务的挖掘效率。

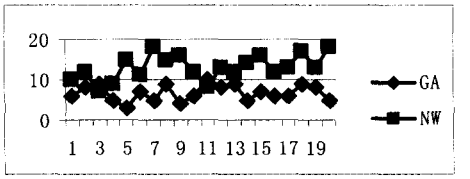

图5 与遗传算法的结果对比

结束语 改进后的算法克服了目前流程挖掘的局限。从实验分析看出本文提出的改进能正确的挖掘不可见任务、弱循环、强循环、跳、非确定性平行结构等复杂结构，但该算法是在完备的日志上进行的挖掘实验，没有考虑噪声的影响。这将是下一步要解决的问题。

参 考 文 献

[1] Cook J E, Wolf A L. Automating process discovery through event-data anal-ysis[C]// Proceedings of the 17th international conference on Software engineering. ACM, New York, NY, USA, 1995: 73-82

[2] Cook J E, Wolf A L. Discovering Models of Software Processes from Event-Based Data[J]. ACM Transactions on Software Engineering and Methodology, 1998, 7(3): 215-249

[3] Huang Hong-mei, Zhang Yun. process mining algorithm base on synchronous-manager. Edition, 2008, 06. 06

[4] Li Jia-fei, Liu Da-you. A process mining algorithm and to discovery duplicate tasks[J]. Journal of Jilin University: Engineering and Technology Edition, 2007, 37(1): 106-110

[5] van der Aalst W M P, van Dongen B F, Herbst J, et al. Weijters. Work◦ow Mining: A Survey of Issues and Approaches[J]. Data and Knowledge Engineering, 2003, 47(2): 237-267

[6] van der Aalst W M P, Weijters A J M M, et al. Process Mining [J]. Special Issue of Computers in Industry, 2004, 53(3)

[7] van der Aalst W M P, Weijters A J M M. Process Mining: A Research Agenda[J]. Computers in Industry, 2004, 53(3): 231-244

[8] Liu Rui, Jiang Ming-hu. The design and implementation of Process mining system based on αalgorithm. 1672-7800(2008) 07-0066-04

[9] www. processmining. com