

# 基于重复任务的复杂结构挖掘

叶小虎<sup>1</sup> 刘芳好<sup>2</sup> 薛岗<sup>2</sup>

(云南大学信息科学与工程学院 昆明 650091)<sup>1</sup> (云南大学国家示范性软件学院 昆明 650091)<sup>2</sup>

**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。在商业流程系统日志中,同名任务和重复任务是大量存在的。现有的挖掘算法都不能很好地区分,这导致在过程挖掘的结果中往往会产生不准确的流程模型。为了提高过程挖掘的准确性,提出了一种改进方法,它不仅能够挖掘日志中的循环结构、非自由选择结构等复杂结构,还能够挖掘日志中的同名任务和重复任务。

**关键词** 过程挖掘,同名任务,重复任务,复杂结构

## Mining Complex Structure Based on the Repetitive Task

YE Xiao-hu<sup>1</sup> LIU Fang-yu<sup>2</sup> XUE Gang<sup>2</sup>

(School of Information Science and Engineering, Yunnan University, Kunming 650091, China)<sup>1</sup>

(National Pilot School of Software, Yunnan University, Kunming 650091, China)<sup>2</sup>

**Abstract** Process mining is helpful for deploying new business processes as well as auditing, analyzing and improving the already enacted ones. The namesake task and a repetitive task exist in the business process of system log of a large number. The existing alpha algorithm of the mining algorithm can distinguish them very well. It leads to the results of the process mining will often produce inaccurate flow model. In order to improve the accuracy of the process mining, it put forward an improved method. The new method not only can dig the complex structure, such as circulation structure, free choice structure and so on, but also can dig the eponymous task and repeat task in the log.

**Keywords** Process mining, Namesake task, Repetitive task, Complex structure

## 1 引言

在信息化高速发展的今天, workflow 技术被广泛应用,其中企业建模是 workflow 技术应用的重要领域。过程挖掘是一种重要的企业建模方法,它可以用来分析、审计、优化现有的系统,帮助改进现有的商务流程,能够帮助企业实现业务流程的建模和再造,提高企业的竞争力。

国内外有很多学者在过程挖掘方面做了大量的工作<sup>[1,2]</sup>。到目前为止,专家们虽然提出了不少有意义、有见解的模型定义语言,例如活动间依赖型模型语言、ADONIS 模型语言、Petri 网模型语言、面向块结构模型语言,它们都有着各自的特点,基本上已经能够很好地完成流程建模。在流程挖掘方面, Agrawal 最早把流程挖掘算法应用到了 workflow 管理系统中。最初,这种算法要求流程中不存在循环,挖掘出的模型用有向无循环图的形式来表示,但 Agrawal 在改进算法中处理了部分的循环问题。Herbst 等人提出基于 ADONIS 模型语言的算法<sup>[7,8]</sup>,该算法能很好地处理流程中含有相同活动名的结构,但它对隐藏结构的发现还没能实现,而且也只能发现简单的循环结构。Aalst 和 Weijter 等人提出了用算法来进行流程挖掘<sup>[10,11]</sup>,这种算法通过对 Petri 网的扩展能发现 And-split 和 And-join 结构的同等替代物,但它对活动间的关系只限于两两关系,因而对类似不自由选择结构 NFC (no-

free-choice) 的类型要考虑全局的结构则不能发现,对于同一流程中含有相同活动名和复杂循环结构这两种情况同样也无法为力。

文献[9]提出了挖掘 workflow 模型的  $\alpha$  算法,并证明了  $\alpha$  算法能准确挖掘的模型种类。因为  $\alpha$  算法采用集合来定义任务之间的关联关系,在判定重复任务时,算法将这些重复任务视为同一个任务,所以往往导致挖掘出错误的过程模型。在文献[3]中提出了一种算法,该算法能较全面地挖掘日志中的不可见任务、循环结构、非确定性平行选择结构等复杂结构,但是因为该算法是在算法的基础上做的改进,所以仍然没能解决同名任务和重复任务的问题。这里将在文献[3]中算法的基础上提出一种改进的方法,使得不仅能够挖掘上述复杂结构,还能挖掘同名任务和重复任务,提高过程挖掘的准确性。

## 2 相关工作

一个任务有时可能在同一 workflow 模型中多次出现。此时,多个任务具有相同的名称,这样的同名任务就是重复任务。重复任务在实际的 workflow 模型中普遍存在。在此首先给出几个必要的定义。

为了确保不会因为日志的丢失导致挖掘结果中结构的丢失,这里使用的日志都是完备日志,为此,下面给出了日志完备性的定义。

本文受云南省教育厅科学研究基础基金项目(2010Y252)资助。

叶小虎(1987-),男,硕士,主要研究方向为流程挖掘, E-mail: xiaohuye@live.cn; 刘芳好(1987-),女,硕士,主要研究方向为网络分布式、workflow; 薛岗(1977-),男,博士,主要研究方向为 workflow 的编制与管理。

**定义 1** 设  $w$  是由任务集  $T$  构成的业务流程产生的事件日志,  $\sigma$  是业务流程一次执行过程产生的任务轨迹,  $\sigma \in W$ .  $W$  具有日志完备性, 当且仅当:

- (1) 对于由任务集  $T$  构成的业务流程产生的任意事件日志  $W'$ , 有  $W' \subseteq W$ .
- (2) 对于任意  $t \in T$ , 都存在  $\sigma \in W$  使得  $t \in \sigma$ .
- (3) 对于事件日志  $W'$ , 若  $W'$  中有  $\sigma_1' = \dots xy \dots, \sigma_2' = \dots xay \dots, \sigma_3' = \dots xaa y \dots$ , 且  $a, x, y \in T$ , 则一定存在  $\sigma_1, \sigma_2, \sigma_3 \in W$  且  $\sigma_1 = \dots xy \dots, \sigma_2 = \dots xay \dots, \sigma_3 = \dots xaa y \dots$ .

循环任务在文献[3]中已经得到了充分的定义, 下面定义重复任务.

**定义 2** 设  $W$  是由任务集  $T$  构成的业务流程产生的事件日志. 当且仅当存在  $\sigma \in W, \sigma = t_1 \dots t_i \dots t_j \dots t_m, t_i = t_j, t_i \notin Loop1(t_i)$  且  $t_i \notin Loop2(t_i)$ , 则称  $t_i$  为重复任务. 重复任务  $t$  的集合记为  $D(t)$ .

重复任务可能出现在 workflow 网中的顺序、并行或选择结构中, 图 1 到图 6 给出了重复任务出现的 6 种组合结构.



图 1 顺序/顺序

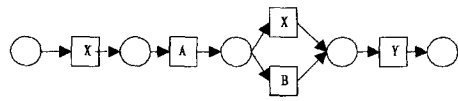


图 2 顺序/并行

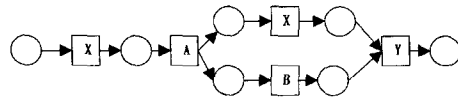


图 3 顺序/选择

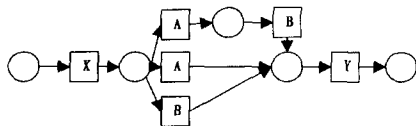


图 4 并行/并行

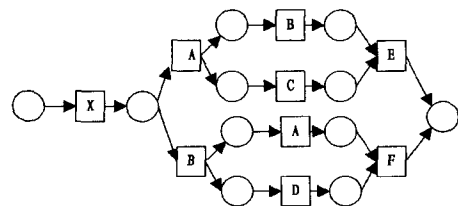


图 5 并行/选择

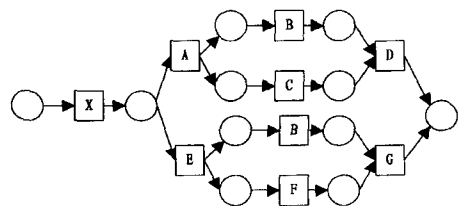


图 6 选择/选择

**定义同名任务:**

**定义 3** 设  $W$  是由任务集  $T$  构成的业务流程产生的事件日志. 当且仅当存在  $\sigma_1, \sigma_2 \in W, \sigma_1 = t_1 \dots t_i \dots t_j \dots t_m, \sigma_2 = t_1 \dots t_i \dots t_j \dots t_m, i \in [1, m]$  且  $j \in [1, n]$ , 若  $t_i, t_j'$  由业务流程中同一位置的同一活动产生, 则称  $t_i, t_j'$  为同一任务. 同一任务  $t$

的集合记为  $S(t)$ .

### 3 基础工作

当业务流程含有循环任务和重复任务时, 产生的事件日志中会出现一个任务多次出现的情况. 为了判定多次出现的任务来自循环任务、重复任务还是同一任务, 需要对任务的结构属性进行判定.

判定同一任务及重复任务.

**属性 1** 设  $w$  是由任务集  $T$  构成的业务流程产生的事件日志. 对于  $\sigma, \sigma' \in W, t, t' \in T, i, j \in [1, n]$ , 若  $t(\sigma, i) = t(\sigma', j)$  则  $\{t(\sigma, i), t(\sigma', j)\} \in S(t)$ .

这个属性容易证明.

**定理 1** 设  $w$  是由任务集  $T$  构成的业务流程产生的事件日志.  $\sigma \in W, t \in T, t \notin Loop2(t)$  且  $len(\sigma) = n (n > 1)$ . 若存在  $i, j \in [1, n], i \neq j, t(\sigma, i) \neq t(\sigma, j)$ , 则  $\{t(\sigma, i), t(\sigma, j)\} \in D(t)$ .

证明(用反证法): 假设  $\{t(\sigma, i), t(\sigma, j)\} \in D(f)$ . 因为  $t(\sigma, i), t(\sigma, j) \notin Loop2(t)$ , 易知  $t(\sigma, i), t(\sigma, j) \in S(t)$ , 因而有  $t(\sigma, i) = t(\sigma, j)$  且  $i = j$ , 这与已知条件  $i \neq j, t(\sigma, i) \neq t(\sigma, j)$  相矛盾, 所以  $\{t(\sigma, i), t(\sigma, j)\} \in D(t)$ . 证毕.

定理 1 保证了在同一个任务轨迹中重复任务的发现.

**定理 2** 设  $W$  是由任务集  $T$  构成的业务流程产生的事件日志.  $\sigma, \sigma', \sigma'' \in W, A, A' \in sub(\sigma) \vee sub(\sigma'), X, Y \in T$ , 则

(1) 若存在  $A \parallel A'$ , 即  $AA' \Theta(\sigma, [X, Y])$  且  $A'A \Theta(\sigma', [X, Y])$ , 则

- $\{A[XAA'Y], A[XA'AY]\} \in S(A)$ ;
- $\{A'[XAA'Y], A[XA'AY]\} \in S(A')$ ;
- $\{X[XAA'Y], X[XA'AY]\} \in S(X)$ ;
- $\{Y[XAA'Y], Y[XA'AY]\} \in S(Y)$ .

(2) 若存在  $A \oplus A'$ , 即  $A \Theta(\sigma, [X, Y])$  且  $A' \Theta(\sigma', [X, Y])$ , 则

- $A[XAY] \in S(A)$ ;
- $A'[XA'Y] \in S(A')$ ;
- $\{X[XAY], X[XA'Y]\} \in S(X)$ ;
- $\{Y[XAY], Y[XA'Y]\} \in S(Y)$ .

(3) 若存在  $A \infty A'$ , 即  $A \Theta(\sigma, [X, Y])$  且  $AA'A \Theta(\sigma', [X, Y])$ , 则

- $\{A[XAY], A[XAA'AY]\} \in S(A)$ ;
- $A'[XAA'AY] \in S(A')$ ;
- $\{X[XAY], X[XAA'AY]\} \in S(X)$ ;
- $\{Y[XAY], Y[XAA'AY]\} \in S(Y)$ .

(4) 若存在  $A \infty$ , 即  $\Theta(\sigma, [X, Y]), A \Theta(\sigma', [X, Y])$  且  $AA \Theta(\sigma', [X, Y])$ , 则

- $\{A[XAY], A[XAAAY]\} \in S(A)$ ;
- $\{X[XAY], X[XAAAY], X[XAAAY]\} \in S(X)$ ;
- $\{Y[XAY], Y[XAAAY], Y[XAAAY]\} \in S(Y)$ .

证明: (1) 由  $\sigma = \dots XAA'Y \dots, \sigma' = \dots XA'AY \dots$ , 得知  $X > A, A > A', A' > Y, X > A', A' > A, A > Y$ . 因为  $A \parallel A'$ , 所以有  $X \rightarrow A, A' \rightarrow Y, X \rightarrow A', A \rightarrow Y$ . 运用改进后算法, 可得 workflow 网  $N_p$ , 从而可得如下结论:

- $\{A[XAA'Y], A[XA'AY]\} \in S(A)$ ;
- $\{A'[XAA'Y], A'[XA'AY]\} \in S(A')$ ;
- $\{X[XAA'Y], X[XA'AY]\} \in S(X)$ ;
- $\{Y[XAA'Y], Y[XA'AY]\} \in S(Y)$ .

(2)-(4)的证明与(1)证明类似。证毕。

**定理 3** 设  $W$  是由任务集  $T$  构成的业务流程产生的事件日志,  $t \in T, \sigma, \sigma' \in W$ 。若在任务轨迹中任务  $t$  和  $t'$  存在关系  $tRt'$  ( $R \in \{>, ||, \oplus, \infty, \infty\}$ ), 在任务轨迹  $\sigma'$  中任务  $t$  和  $t'$  存在关系  $[tR't']$  ( $R' \in \{>, ||, \oplus, \infty, \infty\}$ ) 且  $R \neq R'$ , 则

$$\{t(\sigma, [tRt']), t(\sigma', [tR't'])\} \in D(t);$$

$$\{t'(\sigma, [tRt']), t'(\sigma', [tR't'])\} \in D(t).$$

证明(用反证法): 假设  $\{t(\sigma, [tRt']), t(\sigma', [tR't'])\} \notin D(t)$  ( $R \neq R'$ ) 且  $\{t'(\sigma, [tRt']), t'(\sigma', [tR't'])\} \notin D(t)$  ( $R \neq R'$ ), 可得  $\{t(\sigma, [tRt']), t(\sigma', [tR't'])\} \in S(t)$ ,  $\{t'(\sigma, [tRt']), t'(\sigma', [tR't'])\} \in S(t)$ , 运用定理 2, 可得  $R=R'$ , 这与题设  $R \neq R'$  相矛盾, 因此可得出如下结论:

$$\{t(\sigma, [tRt']), t(\sigma', [tR't'])\} \in D(t);$$

$$\{t'(\sigma, [tRt']), t'(\sigma', [tR't'])\} \in D(t).$$

证毕。

业务流程中的循环任务和重复任务是极其容易混淆的, 但是又确实是分别存在的两种结构, 为了挖掘出正确的流程模型, 必须要仔细的判别。

## 4 算法

在这里把过程挖掘分为预处理、处理和后期处理 3 步。在预处理阶段主要过滤重复任务和同名任务。在处理阶段主要分析日志集合中的循环、非自由平行选择、不可见任务等复杂结构, 并挖掘出初步的工作流模型。在后期处理阶段主要还原更名后的任务的原名称并改变关系集合里其连接库所的连接关系。

基于前面的属性 1 和定理 1-定理 3, 结合文献[3]中的改进算法, 提出了能够挖掘重复任务的另一种改进算法。其中  $T$  为任务集合,  $W$  是任务  $T$  上的 workflow 日志,  $\sigma$  以及任务间的偏序关系参见文献[3]中的定义。具体的算法描述如下。

第一步 过滤重复任务和同名任务。

1) 根据 workflow 日志  $W$  计算出在  $W$  中出现的所有任务构成集合  $D(t)$ ;

2) 运用属性 1 和定理 1-定理 3, 识别出同一任务和重复任务, 并将重复任务  $t$  加入  $D(t)$ ;

3) 对  $D(t)$  中的任务通过添加下标重新命名, 更新重新命名后的日志, 得到新的事件日志  $T_{log}$ 。

第二步 运用文献[3]中新的挖掘算法对  $T_{log}$  进行挖掘, 将最终结果的库所集合命名为  $W^{DT}$ 。

1)  $L1L = \{t \in T_{log} \mid \exists \sigma = e_1, e_2, \dots, e_i, \dots, e_n \in W[e_i \cdot task = e_j \cdot task = e_k \cdot task = t \wedge l = k + 1 = j + 2 = j + 3]\}$ ;

2)  $T' = T_w \mid L1L, L2L = \Phi; F_{LL} = \emptyset$ ;

3) For each  $t \in L1L$  do

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

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

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

4)  $W^{-LL} = \Phi$ ;

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

1.  $\sigma' = \sigma$ ;

2. For each  $t \in L1L$  do  $\sigma' = \text{remove Task}(\sigma', t)$ ,  $T_w = T_w$

$-t$ ;

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

4.  $W^{-LL} = W^{-LL} \cup \sigma'$ ;

6)  $W = W^{-LL}$ ;

7)  $X_w = \{\langle ps, ss, op \rangle \mid ps \subseteq T_w \wedge ss \subseteq T_w \wedge op \in op \wedge \forall a \in ps \forall b \in ss, a \rightarrow_w b, op = \text{default}\}$ ;

8)  $D_s = \{(t_{(P_{in}, P_{out})}; t_{(P'_{in}, P'_{out})}) \mid (P_{in}; P_{out}); (P'_{in}; P'_{out}) \in Y_I \cup Y' \wedge P_{out} \cap P'_{in} \neq \Phi\} \cup \{(a, t_{(P_{in}, P_{out})}) \mid (P_{in}; P_{out}) \in Y_I \cup Y' \wedge \exists (A; X) \in P_{in}; a \in A\} \cup \{(t_{(P_{in}, P_{out})}; b) \mid (P_{in}; P_{out}) \in Y_I \cup Y' \wedge \exists (Y; B) \in P_{out}; b \in B\}$ ;

9)  $D_p = \{(t_{(P_{in}, P_{out})}; t_{(P'_{in}, P'_{out})}) \mid (P_{in}; P_{out}); (P'_{in}; P'_{out}) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; (A'; X') \in P'_{in}; \exists a \in A; a' \in A'; x \in X; x' \in X'; a \parallel_w a' \vee x \parallel_w x'\} \cup \{(t; t_{(P_{in}, P_{out})}) \mid t \in T_w \wedge (P_{in}; P_{out}) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; \exists a \in A; x \in X; a \parallel_w t \vee x \parallel_w t\} \cup \{(t; t_{(P_{in}, P_{out})}) \mid t \in T_w \wedge (P_{in}; P_{out}) \in Y_I \cup Y' \wedge \forall (A, X) \in P_{in}; \exists a \in A; x \in X; a \parallel_w t \vee x \parallel_w t\}$ ;

10)  $T_w = T_w \cup \{(t_{(P_{in}, P_{out})}) \mid (P_{in}; P_{out}) \in Y_I \cup Y'\}$ ;

11)  $(W^{DT}) = (T_w; T_I; T_O; D_s; D_p)$ 。

具体的算法介绍参见文献[3]中的算法说明。

第三步 挖掘后期处理。

1) 将  $T_w$  中重新命名的重复任务的名称恢复到它们原来的名称然后保存到  $T_w$  中;

2) 将  $D_p$  中那些与重新命名的重复任务有关的那些弧连到具有原来的名称的任务上然后保存到  $D_p$ ;

3) 置  $N = (W^{DT}) = (T_w; T_I; T_O; D_s; D_p)$ 。

算法结束。

## 5 实验

实验中用到的算法的源程序采用 C 语言编写, 实现了包括文献[3]中算法在内的实验原型。实验中采用 CPN Tools (color petry net tools) 建立了 5 个含有重复任务的工作流模型。实验时, 首先为每个模型随机产生一个包含 1000 个事件轨迹的工作流日志, 接着对日志进行分析。图 7 给出了实验原型和采用改进算法挖掘的模型结果的对比。因为篇幅原因, 这里只给出了一组模型对比。通过实验对比, 改进算法能识别出重复任务, 挖掘出与原来模型一致的工作流模型。

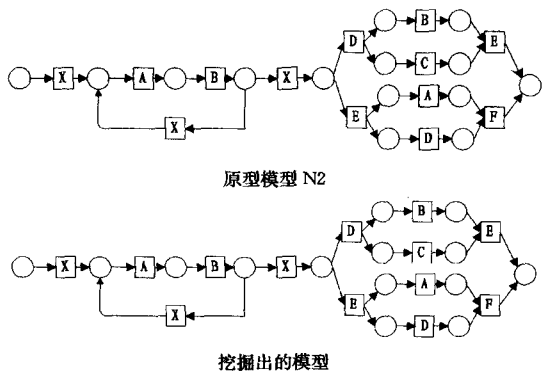


图 7 模型对比

实验结果表明改进后的算法对日志中的同名任务和重复任务是可以有效区分、挖掘的。

根据改进后的算法编写 Prom 的算法插件, 然后选取 5 组例子日志作为实验日志, 运行改进后的挖掘算法得到的实验结果如图 8 所示。

(下转第 224 页)

义新的 URL 请求变量

```
fileRefs.browse([new FileFilter("所有文件(*.*)", "*.*")]); //浏览本地资源管理器,选择需要上传的文件
for each (var f; FileReference in fileRefs.fileList)
selectedFiles.addItem(f); //将每个选择的文件填充至 filereferencelist 类
for each (var f; FileReference in selectedFiles)
{try{f.upload(urlrequest);} //逐个上传文件
catch (e; Error){Alert.show(e.message);} //捕获异常
}
```

其中,连接数据库的字符串为:

```
connectionString1="data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|\aspnetdb.mdf;User Instance=true"; //数据源为 SQL Server 2005;
connectionString2="Server=DiskSystemServerAdmin;Database=SK_Disk;User ID=DiskSystem;Password=disksystem" //数据库服务器名称、数据库名称、用户名和密码
```

**结束语** 网络信息技术的不断发展使得企业和个人的信息数据量持续增加,通过建立基于 Web 的网络硬盘系统为不断增长的数据提供存储和备份空间,并满足异地存取的需求,系统的应用验证了中小企业部署此种方案的可行性。通过测试发现最大本地上传速度达 10MB/s,异地最大上传速度达 500KB/s,基本满足日常文件存储的速度要求。考虑到系统采用 IIS 应用服务,特别是在目前网络漏洞频繁出现、威胁和木马日益猖獗的情况下,如何提高系统自身安全性,保护用户

信息和数据的安全将是下一步研究的重点。

## 参考文献

- [1] 张迪,朱立谷,等.基于 WEB 的移动端云存储技术研究[J].计算机工程与应用,2010,46(36):66-67
- [2] 朱率率,杨晓元,等.动态自适应网络存储系统的设计与实现[J].计算机应用研究,2010,27(8):3132-3134
- [3] 王施人,陕振,等.基于 iSCSI 存储集群的设计与实现[J].计算机工程与设计,2010,31(11):2598-2560
- [4] 姚全珠,李宏涛.改善 iSCSI 存储系统性能优化策略[J].计算机工程与设计,2010,31(15):3419-3420
- [5] 边根庆,高松,等.面向分散式存储的云存储安全架构[J].西安交通大学学报,2011,45(4):41-43
- [6] 熊爱金,郝玉洁.基于 iSCSI 协议的 IP 存储研究及网络存储技术发展现状[J].软件导刊,2009,8(5):146:148
- [7] 杨扬,贾君君,李为卫.基于 J2EE 办公自动化系统的应用研究[J].计算机与现代化,2010,180(8):185-186
- [8] 崔宝娟,苏中滨,沈维政.基于 ASP.NET 的网站群动态建站研究[J].计算机应用与软件,2011,28(3):116-117
- [9] 周益宏,陈建勋.浅析基于 ASP.NET 的网站 SQL 注入攻击及防范措施[J].计算机安全,2010,12(6):93-94
- [10] 唐力,槐寅,陈震.Flash 媒体服务器的优化部署[J].清华大学学报:自然科学版,2010,50(1):6-8
- [11] 顾兵. SQL Server 2005 中的 XQuery 应用研究[J].计算机技术与发展,2011,21(3):98-99

(上接第 211 页)

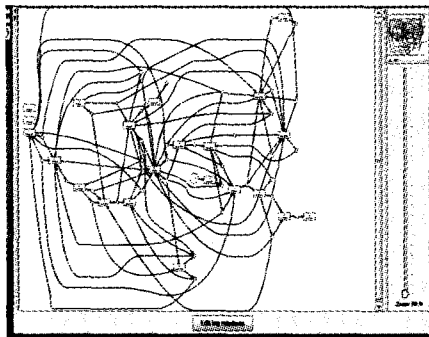


图 8 挖掘结果

实验表明改进后的算法对于不可见任务、循环结构、非自由选择结构仍然是能够正确挖掘的。

**结束语** 改进后的算法克服了算法挖掘的局限。从实验分析看出本文提出的改进能正确地挖掘系统日志中的同名任务和重复任务,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,仍然是能够正确挖掘,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,且复杂结构挖掘的准确率受到影响,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 analysis[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] Ye Xiao-hu, Yao Shao-wen, Liu Fang-yu, et al. Towards Improving Process Mining Results Base on Invisible Tasks[Z]. ACM-SIC2011 B0406155
- [4] Van der Aalst W M P, Weijters A J M M. Process mining—a research agenda[J]. Computers in industry, 2004, 53(3):23-244
- [5] Agrawal R, Gunopulos D, Leymann E. Mining process models from workflow log[C]//Proceedings of the Sixth International Conference on Extending Database Technology. USA, 1998: 469-483
- [6] Herbst J, Dealing with concurrency in workflow induction[C]//Proceedings if the 7 European Concurrent Engineering Conference. SCS Europe, 2000
- [7] Joachim H, Karagiannis D. Workflow mining with InWoLvE [J]. Computers in Industry, 2004, 53(4):245-264
- [8] Schimm G. Mining exact models of concurrent workflows[J]. Computers in Industry, 2004, 53(2):265-281
- [9] Aalst W M P, Weijters T, Maruster I. Workflow mining: discovering process models from event logs[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9):1128-1142
- [10] Van der Aalst W M P, Weijters A J M M, Marudter L. Workflow mining, which processes Can be Rediscovered [C]//IEEE Transactions on Knowledge and Data Engineering. Eindhoven, 2002:67-91
- [11] Van der Aalst W M P, Van Dongen B E. Discovering workflow performance models from timed logs[C]//International Conference on Engineering and Deployment of Cooperative Information System. Berlin, 2002:45-63