経営課題にAIを!ビジネスインフォマティクス研究会 (第18回)
JSAI Special Interest Group on Business Informatics (SIG-BI #18)

作業順序を考慮した作業者の代表的な活動パターンの抽出方法に関する考察

A Study on Extracting Workers' Representative Activity Patterns by Considering the Work Order

宮森望 [1*]　堀田大貴 [1]　上田賀一 [1]
Nozomi Miyamori[1]　Hiroki Horita[1]　Yoshikazu Ueda[1]

[1] 茨城大学
[1] Ibaraki University

Abstract: Companies are now able to gather a lot of business process's execution data (called event log) by managing them with information systems. In addition, they seek to analyze their process by leveraging event log to improve their processes (e.g. reducing waiting time and costs). However, there are many cases where process management systems record work suspending, but do not record specific reason for that suspending. Hence, we propose the extracting representative workers' activity patterns method for understanding the activity of workers during work suspending. The method takes into account the order of work with Damerau – Levenshtein distance. Analysts are able to understand the cause of the work stoppage by comparing the extracted patterns with the process flow. We could find the workers' activity patterns which is along work flow of the process in the empirical evaluation.

1 はじめに

1.1 背景

企業の製品製造やサービス提供のための業務活動全体をビジネスプロセス (または単にプロセス) と呼び, また WFM, CRM, SCM などの情報システムが普及したことで情報システムを用いたプロセスの管理とプロセスの実行データの収集が行われるようになった. プロセスの実行データをイベントログと呼び, イベントログを活用したプロセスの分析手法としてプロセスマイニング [Aalst 16, Aalst 12] が注目されている.

1.2 本研究の動機

プロセスの開始から終了までの時間を削減することは迅速な製品製造やサービス提供につながるため重要であり, また, 待ち時間[1]はプロセスの実行時間の多くを占めるとされている [Dumas 18]. そのため待ち時間の削減はプロセスの改善に効果的であると考えられる. 表 1 は, 図 1 で示した作業を 4 分類し, 分類した作業が占める時間の割合である. ここでは作業者の割当と作業一時停止の時間が待ち時間に相当する. 全体の中でも作業の停止に伴う待ち時間が多くの時間を占めている. とりわけ理由の明記されていない作業の一時停止は全体の約 43 %を占めている.

表 1: プロセスの全時間に占める作業の割合

作業種	割合
作業者の割当	27.9 %
実務	9.6 %
作業一時停止	19.4 %
作業一時停止 (理由が不明確)	43.1 %

作業者による作業の一時停止は待ち時間が発生する直接的な原因であるが作業停止の理由について明記されていない事例も存在する. 明記されない理由として, 例えば, 一時停止の理由の選択肢がシステムに定義されていないこと, 自由記述の場合でも理由を明記することに時間を費やすよりも実務の時間に費すことなどが考えられる.

待ち時間の削減には待ちを発生させている活動に対して対策を取る必要がある. しかしながら一時停止の具体的な理由が不明である, もしくは一時停止を起こしている活動に対する理解が浅いと対策を打つことが

*E-mail: 19nm730f@vc.ibaraki.ac.jp
[1]プロセスを進めるための実務が行われていない時間を指す

困難であると考えられる．そこで本研究ではイベントログから作業者の活動パターンを抽出することで作業停止の発生原因の分析を試みた．具体的にはイベントログから作業停止が発生している間の作業者の活動のみを抽出したデータ (サスペンドログと呼ぶ) を構成し, サスペンドログから作業順序を考慮した作業者の代表的な活動パターンを抽出する．ここでの代表的という言葉は全体の中で類似かつ頻出していることを指す．作業順序を考慮するためにダメラウ/レーベンシュタイン距離による類似度を用いる．実データを用いた実験では得られた活動パターンをプロセスモデルと比較することで作業者の活動の理解を試みた．

1.3 本論文の構成

以降の本論文の構成は次の通りである．2 節では本研究の関連研究であるプロセスディスカバリとビジネスプロセスの時間に関する研究について述べる．3 節では背景知識としてイベントログの定義を与え, また作業順序を考慮した作業者の活動の類似度を説明するために編集距離を導入する．4 節では提案手法について説明する．5 節では実プロセスのデータセットに対して行った実験について述べる．6 節では本研究のまとめと今後の課題を述べる．

2 関連研究

プロセスマイニングにはプロセスディスカバリと呼ぶ手法がある [Aalst 16]．プロセスディスカバリではイベントログからプロセスを発見することを目的とし, 具体的にはプロセスを未知の状態遷移機械 (オートマトンやペトリネットなど) とみなし, 与えられたイベントログを生成したプロセスと同値な状態遷移機械を導出する．プロセスディスカバリの著名な研究に Aalst らによる α アルゴリズムがある [Aalst 16]．α アルゴリズムでは入力にイベントログが与えられ, 出力にペトリネットで表現されたプロセスを出す．アルゴリズムの内容はただ 1 つのプロセスの開始状態とただ 1 つの終了状態が存在すると仮定し, イベントログに現れた作業の並びを事前に定義されている 3 つの遷移規則に当てはめることで開始から終了までの区間の遷移を求めている．また α アルゴリズムが持つ問題点を解消することを目指した別のプロセスディスカバリの研究として [Weijters 11] の研究がある．[Weijters 11] では α アルゴリズムでは表現できない閉路を含んだプロセスの導出や, イベントログに現れる作業の並びの頻度を考慮することによりプロセスの主要な流れのみの抽出が行える．

プロセスディスカバリではプロセスモデルの発見が目的であることに対し, 本研究ではプロセスの作業者の活動パターンの発見を目指した．またプロセスディスカバリアルゴリズムはプロセスモデルの発見を目指しておりそのまま作業者の活動パターンの発見に応用することは難しい．一方でプロセスに属する作業者同士の関連を分析する手法として [Song 08] がある．[Song 08] では作業者間の作業遷移の記述方法や, プロセスでの作業者の役割を発見する手法などが提案されている．[Song 08] と比較して作業者に着目している点は本研究と同一である一方で, 本研究では作業者の活動パターンを発見することに主眼が置かれている．

ビジネスプロセスの時間に関する研究としてプロセスのタスクの完了時間やケースの完了時間の予測の研究がある [Aalst 11, Senderovich 15, Verenich 17]．[Aalst 11] ではプロセスモデル上のアクティビティにイベントログから得られる時間に関する情報を付与することでプロセスの実行時間をシミュレートしタスクやケースの完了時間の予測を行っている．[Senderovich 15] ではプロセス中の待ち作業とサービス提供の作業を定義し, それぞれを待ち行列のモデルに当てはめることによりケースの完了時間を予測している．また [Verenich 17] では作業間の遷移の仕方に 4 つのルールとその遷移時間の定義式を用意し, 分析したいプロセスモデル中の遷移をルールにあてはめることでケースの完了時間の予測を行っている．

ビジネスプロセスの遅れの原因分析を行った研究として [Hompes 17] がある．[Hompes 17] ではイベントログの属性の時間的な変化に着目することで, 最終的な遅れと相関の強い属性を抽出することで遅れの原因分析を行っている．

[Denisov 18] ではあるアクティビティから別のアクティビティへの遷移の仕方の時間的な変化を視覚化する方法を提案している．[Denisov 18] の手法ではあるアクティビティから別のアクティビティへの遷移がバッチ処理的に行われているのか, 逐次行われているのかといった様子を視覚化できる．

3 背景知識

3.1 表記法について

本節では本論文で用いる数学的表記法について説明する．

集合 A から集合 B への関数 f を

$$f \in A \to B$$

と書く．

表 2: イベントログの例

Case	Activity	AID	Worker	Time
1	In Progress | end	1	a	11:00
	Suspend | start	2	a	11:00
	Suspend | end	2	a	16:10
2	In Progress | start	4	a	11:10
	In Progress | end	4	a	13:00
	Suspend | start	5	a	13:10
	Suspend | end	5	a	14:30
	In Progress | start	6	a	15:00
	In Progress | end	6	a	16:00

集合 A が与えられたとき A^* を A の要素による有限列の全体集合を表す. ある有限列 $S \in A^*$ を与えたとき S の i 番目の要素を a_i と書く. 有限列 S が n 個の要素からなる場合

$$S = (a_1, a_2, \cdots, a_{n-1}, a_n)$$

と書く. また有限列 S が要素 a_i を含むことを

$$a_i \in S$$

と書く.

関数 $f \in A \to B$ と関数 $f_{A^*} \in A^* \to B^*$ が与えられたとき有限列 $S \in A^*$ に関数 f_{A^*} を適用して得られる値を

$$f_{A^*}(S) = (f(a_1), f(a_2), \cdots, f(a_{n-1}), f(a_n))$$

と書く代わりに表記を簡略化して

$$f(S) = (f(a_1), f(a_2), \cdots, f(a_{n-1}), f(a_n))$$

と表記する.

3.2 イベントログ

イベントログは表 2 のような形式で表されることが多い. それぞれの行をイベントと呼ぶ. 列を属性と呼び, それぞれのイベントはある属性に関して属性値を持つ. 属性と属性値の例として表 2 は 2 列目の Activity が属性であり, 1 行 2 列目の値 In Progress | end が属性値である. また水平線で区切られたまとまりをケースと呼ぶ. ケースはそれぞれがプロセスの開始から終了までの 1 つの実行記録を表す. またイベントは必ず 1 つのケースにのみ属する (同じイベントが 2 つ以上のケースに存在しない).

以下ではイベント, 属性, ケースの定義ををもとにイベントログの定義を与える.

定義 1. (イベント, 属性) 全体集合 $\mathcal{E}$ が与えられ要素 $e \in \mathcal{E}$ をイベントと呼ぶ. イベントの有限集合 $E \subset \mathcal{E}$ から全体集合 U の部分集合 $V \subset U$ への関数 $\alpha \in E \to V$ を属性と呼び, 集合 U の要素 $u \in U$ を属性値と呼ぶ. あるイベント e が属性 α の属性値 v を持つことを

$$\alpha(e) = v$$

と書く.

定義 2. (ケース) イベント $e \in E$ と属性 $C \in E \to V$ が与えられイベント e の属性 C の値が $C(e) = c_i$ であるというとき e はケース c_i に属するといい

$$e \in c_i$$

と書く. また, イベント e がケース c_i に属するときイベント e は c_i を除くケース x には属さない. すなわち

$$e \in c_i \iff e \notin x \wedge x \neq c_i$$

を満たす.

定義 3. (イベントログ) $L = (E, A \cup \{C\})$ の 2 つ組が与えられたとき L をイベントログと呼ぶ. ここで $E \subset \mathcal{E}$ はイベントの有限集合, $A = \{\alpha \mid \alpha \in E \to V\}$ は属性の有限集合, C はケース属性を表す.

以下ではイベント e_i がイベントログ E に含まれるとき

$$e_i \in E$$

と書く. 同様にケース c_i がイベントログに含まれるとき, すなわち

$$\exists e(e \in E \wedge e \in c_i)$$

を満たすとき, 表記を簡略化して

$$c_i \in E$$

と書く. 4.1 節で導入するサスペンドログとサスペンドインスタンスに関しても同様の表記を用いる.

表 2 のイベントログの例を用いてイベント, 属性, ケースを表記する. 1 行目のイベントは e_1 と表され同イベントの Activity 属性の値は $ACT(e_1) =$ In Progress | end となる. また 1 番目のケースを $c_1 = (e_1, e_2, e_3)$ と書く.

また提案手法で用いる属性として以下の属性がある (カッコ内は数式中での表記).

Activty (ACT): 作業名

AID (AID): 作業の開始と終了を対応づけるための作業の ID

Worker (W): 作業者名

Time (T): イベントが記録された時刻

3.3 編集距離

提案手法において作業者の活動の類似度を測るためにダメラウ/レーベンシュタイン距離 [Brill 00] を導入する. ダメラウ/レーベンシュタイン距離は 4 つの操作, 要素の挿入, 削除, 置換, 隣接交換で定義される. それぞれ例を用いて説明する.

1 回の挿入とは有限列 $(a, b) \in A^*$ と要素 $c \in A$ が与えられたとき以下の 3 つの有限列

$$(c, a, b) \quad (a, c, b) \quad (a, b, c)$$

を得る操作である.

1 回の削除とは有限列 (a, b) が与えられたとき以下の 2 つの有限列

$$(a) \quad (b)$$

を得る操作である.

1 回の置換とは有限列 $(a, b) \in A^*$ と要素 $c \in A$ が与えられたとき以下の 2 つの有限列

$$(c, b) \quad (a, c)$$

を得る操作である.

1 回の隣接交換とは有限列 (a, b) が与えられたとき以下の 1 つの有限列

$$(b, a)$$

を得る操作である.

以上より, 編集距離とは 2 つの有限列 S と T が与えられ有限列 S を上記のいずれかの操作を繰り返し適用し T を得たときの操作回数の総計を指す.

また, 2 つの有限列 S と T のダメラウ/レーベンシュタイン距離にもとづく類似度は

$$\frac{d(S, T)}{max(|S|, |T|)}$$

で与えられる. ここで $d(S, T)$ は S と T のダメラウ/レーベンシュタイン距離, $|S|$, $|T|$ はそれぞれ S と T の列の長さを表す.

4 提案手法

提案手法は大きく 2 つのステップから成る. まずイベントログから作業停止が発生している間の作業者の活動のみを抽出したデータ であるサスペンドログを構成し, 次にサスペンドログから作業順序を考慮した作業者の代表的な活動パターンを抽出する. 以下では, まずサスペンドログについて説明し, 続いて代表的な活動パターンの抽出法について説明する.

表 3: 表 2 のイベントログから得られるサスペンドログ

SI	Activity	AID	Worker	Time
1	Suspend | start	2	a	11:00
	In Progress | start	4	a	11:10
	In Progress | end	4	a	13:00
	Suspend | start	5	a	13:10
	Suspend | end	5	a	14:30
	In Progress | start	7	a	15:00
	In Progress | end	7	a	16:00
	Suspend | end	2	a	16:10
2	Suspend | start	5	a	13:10
	Suspend | end	5	a	14:30

4.1 サスペンドログ

本節では待ちが発生中 (作業を一時停止中) の作業者の活動データであるサスペンドログを導入する. 表 3 は表 2 のイベントログから得られるサスペンドログの例である. サスペンドログはイベントログのケースのようにサスペンドインスタンスという属性を持つ. 表 3 における最左列 SI がサスペンドインスタンスの ID を表す.

また, すべてのサスペンドインスタンスは次の条件をみたす.

- インスタンスの最初のイベントのアクティビティの値は Suspend | start
- インスタンスの最後のイベントのアクティビティの値は Suspend | end
- インスタンスの最初のイベントのアクティビティ ID と最後のアクティビティ ID が同じ
- インスタンスに含まれるイベントの時刻は最初のイベントの時刻から最後のイベントの時刻の範囲内の値をとる
- インスタンス内の作業者はそのインスタンスの 1 番目の作業者と同じ値をとる

以上をもとにサスペンドログの形式的な定義を与える.

定義 4. (サスペンドログ) イベントログ L が与えられたときイベントログから得られる 2 つ組 $SL = (E_{SL}, A_{SL} \cup \{SI\})$ をサスペンドログと呼ぶ. ここで $E_{SL} \subset E$ はイベントの部分集合, $A_{SL} \subset A$ は属性の部分集合, SI はサスペンドインスタンス属性を表す. また全てのサスペンドインスタンスに関して次を満たす.

- $ACT(s_1) =$ Suspend | start
- $ACT(s_n) =$ Suspend | end
- $AID(s_1) = AID(s_n)$

- $T(s_1) < T(s_i) < T(s_n) \quad (1 < i < n)$
- $W(s_1) = W(s_i) \quad (1 \leq i \leq n)$

ここで $ACT \in A_{SL}$ はアクティビティ属性, $AID \in A_{SL}$ はアクティビティ ID 属性, $T \in A_{SL}$ はタイムスタンプ属性, $W \in A_{SL}$ は作業者属性, n はサスペンドインスタンスの長さ, s_i はサスペンドインスタンスに属する i 番目のイベントを表す.

4.2 代表的な活動パターンの抽出

本節ではサスペンドログから作業者の代表的な活動パターンの抽出方法について説明する. サスペンドインスタンス $SI_i = (e_1, e_2, \cdots, e_{n-1}, e_n)$ から得られる以下のアクティビティ列

$$\begin{aligned} &ACT(SI_i) \\ &= (ACT(e_1), ACT(e_2), \cdots, ACT(e_{n-1}), ACT(e_n)) \end{aligned}$$

を作業者の活動パターンと呼ぶ. 以下ではサスペンドログ中の全てのサスペンドインスタンスから得られる活動パターンの集合を $ACT(SL) = \{ACT(SI_i) \mid SI_i \in SL\}$ と書く.

サスペンドログから得る作業者の活動パターンの集合 $ACT(SL)$ は含まれる作業者の活動の数が膨大になるためそのまま用いた待ち時間の原因の分析は現実的ではない. そこで提案手法では活動パターン $ACT(SL)$ から代表的な活動を抽出することでこの問題に対処する.

アルゴリズム 1 に代表的な活動パターンの抽出方法を示す.

アルゴリズム 1 作業者の代表的な活動パターンの抽出

Require: $ACT(SL)$ ▷ 作業者の活動パターンの集合
Require: M ▷ 活動の類似度行列
Require: C ▷ 活動の出現回数
Require: θ ▷ 活動の類似度の閾値
  $W \leftarrow \mathrm{MultiplyEachRows}(M, C)$
  $m \leftarrow \mathrm{MaxEachColumns}(W)$
  $W \leftarrow \mathrm{DivideEachColumns}(W, m)$
  $I \leftarrow \{\phi\}$
  while $\mathrm{Mean}(W) \leq \theta$ **do**
    $\mu \leftarrow \mathrm{MeanEachRows}(W)$
    $i \leftarrow \mathrm{argmin}(\mu)$
    $W \leftarrow W_{ii}$ ▷ i 行 i 列を除いた小行列
    $I \leftarrow I \cup \{i\}$
  end while
  return $ACT(SL) \setminus \{ACT(SI_i) \mid i \in I\}$

以下では例を用いてアルゴリズムを説明する.

活動の類似度行列 M において M の要素 m_{ij} は作業者の活動パターンの要素 $s_i, s_j \in ACT(SL)$ 間の類似度を表す. 例として $ACT(SL) = s_1, s_2, s_3$ が与えられその類似度行列 M が

$$M = \begin{pmatrix} 1 & 0.8 & 0.7 \\ 0.8 & 1 & 0.6 \\ 0.7 & 0.6 & 1 \end{pmatrix}$$

として与えらたとき類似度 $m_{12} = 0.8$ は s_1 と s_2 の類似度を表す. 本論文では類似度として 3.3 節で説明したダメラウ/レーベンシュタイン距離にもとづく類似度を用いる. ダメラウ/レーベンシュタイン距離を用いることで作業順序を考慮した類似の作業者の活動パターンを抽出できると期待できる.

続いて類似度行列 M と作業者の活動の出現回数 C を用いて, 重み付きの類似度行列 W を構成する. 最終的な W の値がそれぞれの列において最大値が 1 となるように正規化する. 例として作業者の活動の出現回数 $C = (2, 2, 1)$ と先例の類似度行列 M が与えられたとき $\mathrm{MultiplyEachRows}(M, C)$ の値は

$$\begin{aligned} &\mathrm{MultiplyEachRows}(M, C) \\ &= \begin{pmatrix} 1 \times 2 & 0.8 \times 2 & 0.7 \times 2 \\ 0.8 \times 2 & 1 \times 2 & 0.6 \times 2 \\ 0.7 \times 1 & 0.6 \times 1 & 1 \times 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 & 1.6 & 1.4 \\ 1.6 & 2 & 1.2 \\ 0.7 & 0.6 & 1 \end{pmatrix} \end{aligned}$$

となる. $W = \mathrm{MultiplyEachRows}(M, C)$ と与えられそのとき $m = \mathrm{MaxEachColumns}(W)$ の値は

$$m = \begin{pmatrix} 2 & 2 & 1.4 \end{pmatrix}$$

となる. W と m を用いて $\mathrm{DivideEachColumns}(W, m)$ を計算すると

$$\begin{aligned} &\mathrm{DivideEachColumns}(W, m) \\ &= \begin{pmatrix} 2/2 & 1.6/2 & 1.4/1.4 \\ 1.6/2 & 2/2 & 1.2/1.4 \\ 0.7/2 & 0.6/2 & 1/1.4 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0.8 & 1 \\ 0.8 & 1 & 0.857 \\ 0.35 & 0.3 & 0.714 \end{pmatrix} \end{aligned}$$

となる.

ループでは重み付き類似度行列 W の全体の平均 $\mathrm{Mean}(W)$ が閾値 θ を超えるまで W から指定の行と列を取り除く. 取り除く行と列は W の行毎の平均で最も小さいものを選ぶ. 例えば活動の類似度の閾値を $\theta = 0.8$ とし

たときループの 1 回目では Mean(W) の値が

$$\begin{aligned}&\mathrm{Mean}(W)\\&=\frac{1+0.8+1+0.8+1+0.857+0.35+0.3+0.714}{9}\\&=0.759\end{aligned}$$

なので $\mu = \mathrm{MeanEachRows}(W)$ の値を計算すると

$$\begin{aligned}\mu &= \begin{pmatrix}(1+0.8+1)/3\\(0.8+1+0.857)/3\\(0.35+0.3+0.714)/3\end{pmatrix}^T\\&=\begin{pmatrix}0.933 & 0.886 & 0.455\end{pmatrix}\end{aligned}$$

となり, 削除する行と列の番号は $\mathrm{argmin}(\mu) = 3$ なので i 行目 i 列目を取り除いた小行列は W_{ii} は

$$W_{ii} = W_{33} = \begin{pmatrix}1 & 0.8\\0.8 & 1\end{pmatrix}$$

となる. 上記の W_{ii} の値で更新した W を用いた 2 回目のループでは Mean(W) の値が

$$\begin{aligned}\mathrm{Mean}(W) &= \frac{1+0.8+0.8+1}{4}\\&=0.9\end{aligned}$$

なので $\mathrm{Mean}(W) = 0.9 \leq 0.8 = \theta$ を満たさなくなりループを抜ける.

最後に重み付き類似度行列 W から取り除いた行と列に対応する作業者の活動を与えられた活動パターン $ACT(SL)$ から取り除くことで得られる集合 $ACT(SL) \setminus \{ACT(SI_i) \mid i \in I\}$ が最終的な出力の活動パターンである. 先述までの例を用いると $I = 3$ なので最終的に得られる活動パターン $ACT(SL) \setminus \{ACT(SI_i) \mid i \in I\}$ の値は

$$\begin{aligned}&ACT(SL) \setminus \{ACT(SI_i) \mid i \in I\}\\&= \{s_1, s_2, s_3\} \setminus \{s_3\}\\&= \{s_1, s_2\}\end{aligned}$$

となる.

5 実験

本節では実プロセスのインシデント管理プロセスを対象に実験を行った (データセットは BPI Challenge 2013 で公開されている[2]). 実験では対象プロセスのイベントログに提案手法を適用し得られた活動パターンと実験対象のプロセスを比較によってどのような活動が行われているか調査した. また提案手法で得られる活動パターンが占める待ち時間の割合と, 活動パターンの種類数についても調査を行った.

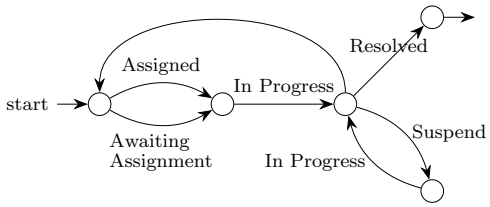


図 1: 実験対象のプロセス

図 1 は実験対象のプロセスモデルである. 初めにインシデントの登録によってケースが開始する. Awaiting Assignment と Assigned は共にインシデントの作業者の割り当てを指す. 両者の違いは Assigned がインシデントを他の作業者に明示的に割り当てたのに対して, Awaiting Assignment はインシデントの割り当ては作業者自身から受け持つ (すなわち Assigned は作業者が事前に特定されているのに対し, Awaitin Assignment の場合は任意の作業者が受け持つことができる). In Progress はインシデント解決のための実務を表す. Suspend は進行中の実務の一時停止を表す[3]. 最終的にインシデントの対応策が承認されると Resolved が発行されケースが終了する. またラベルのない遷移はダミーの遷移でありイベントログには記録されない.

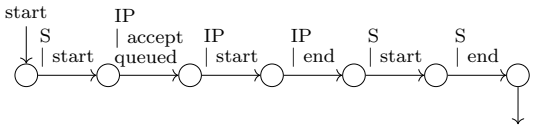
実験対象のイベントログにおいて Suspend と In Progress はそれぞれ開始と終了を区別するために start と end の接尾辞を持つ (例えば Suspend の開始は Suspend |start のように書く). また In Progress は Assigned と Awaiting Assignment のどちらから開始したかを表す accept assigned と accept queued の接尾辞を持つ.

実験対象のイベントログに含まれるイベントの数は 71307, ケース数は 6126 であり, 同イベントログから得たサスペンドログに含まれるサスペンドインスタンスの数は 2707, そのうち長さ 3 以上のサスペンドインスタンスの数は 1618 であった. 以下で用いるサスペンドログは長さ 3 以上のサスペンドインスタンスのみを含む.

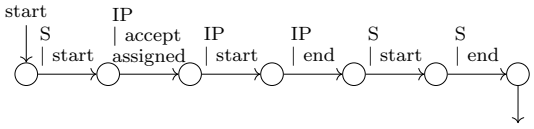
図 2 はアルゴリズム 1 の活動の類似度の閾値を $\theta = 0.6$ としたときに得られた作業者の活動パターンである. 図中では S は Suspend の略であり, IP は In Progress の略である. 得られた活動パターンでは図 2a, 2b に共通する点として作業者は作業の一時停止の後, 登録されているインシデントを受理している. その後は共に共通の流れである. この活動パターンはインシデントの受理後, 開始してからある程度まで作業を続け, 進めた

[2] https://www.win.tue.nl/bpi/doku.php?id=2013:challenge

[3] 実際のデータでは Wait - User というラベルが与えられている



(a) 活動パターン 1



(b) 活動パターン 2

図 2: 提案手法を適用し抽出した作業者の活動パターン

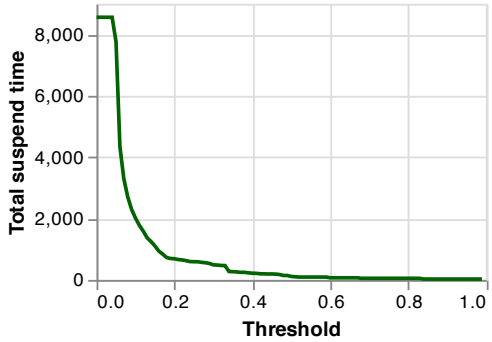


図 3: 閾値 θ に対する抽出した活動パターンの総待ち時間

作業を一旦中断し, 最初に中断していた作業を再開する流れと解釈できる．また得られた活動パターンは作業の順序がプロセスの規定する作業の順序と一致していることが見て取れる．

続いてアルゴリズム 1 を適用して得られる活動パターンが全体の待ち時間をどれだけ占めるかの実験結果を示す．図 3 はアルゴリズム 1 の活動の類似度の閾値 θ を変化させたときの活動パターンが占める待ち時間である．閾値 $\theta = 0$ のとき抽出される活動パターンの数が最大となり全体の待ち時間と一致する．提案手法において活動の類似度の閾値 θ を大きくすると全体の待ち時間に対して得られる活動のパターンの待ち時間の割合も大きく減少することが結果となった．

最後にサスペンドログから得られる作業者の活動パターンの種類数についての実験結果を示す．図 4 は階層クラスタリング (WPGMA) [Aggarwal 15] を適用し得られたクラスタの数の推移である．活動間の距離はダメラウ/レーベンシュタイン距離を用いた．また表 4 は図 4 のデータの一部である．クラスタ間の距離 (クラスタ内の要素間の最大距離から 1 足したもの) を大きくすると急激にクラスタ数が減少することが見て取れる．しかしながらクラスタリングによって得られた活動パターンをプロセスの分析者が目視で比較する場合を考慮すると表 4 のデータからクラスタ間の距離とクラスタ数ともに十分に小さな数に収まっていないと考えられる．

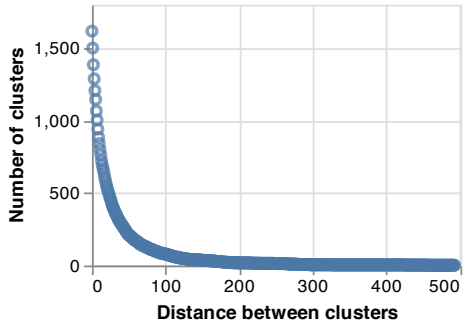


図 4: クラスタ間距離に対するクラスタ数

表 4: 図 4 に示すデータの具体的な値

クラスタ間距離	クラスタ数
1	1498
5	1148
10	841
50	212
100	79
200	18
300	5

6 まとめ

本研究ではイベントログから作業者の活動パターンを抽出し作業停止中の作業者の活動の理解を試みた．具体的にはイベントログから作業停止が発生している間の作業者の活動のみを抽出したサスペンドログと呼ぶデータを構成し, サスペンドログから作業順序を考慮した作業者の代表的な活動パターンを抽出する手法を提案した．ここでの代表的という言葉は全体の中で類似かつ頻出していることを指す．作業順序の考慮にはダメラウ/レーベンシュタイン距離による類似度を用いた．

実データを用いた実験では得られた活動パターンをプロセスモデルと比較することで作業者の活動の理解を試みた．得られた活動パターンは作業の順序がプロセスの規定する作業の順序と一致していることが確認できた．しかしながら提案手法によって得られる作業者の代表的な活動パターンは全体の待ち時間のうち一部のみを占める結果となった．

今後の課題として今回の手法では網羅できなかった部分の作業者の活動パターンとその部分の待ち時間の分析が必要である．またアクティビティ列と編集距離にもとづく作業者の活動パターンの抽出では活動パターン数が多くなりすぎる問題が生じた．この問題の対処にはアクティビティ列と編集距離とは異なる指標を用いた活動パターンの抽出が考えられる．

参考文献

[Aalst 11] Aalst, van der W., Schonenberg, M., and Song, M.: Time prediction based on process mining, *Information Systems*, Vol. 36, No. 2, pp. 450–475 (2011), Special Issue: Semantic Integration of Data, Multimedia, and Services

[Aalst 12] Aalst, van der W., et al.: Process Mining Manifesto, in *Business Process Management Workshops*, pp. 169–194, Berlin, Heidelberg (2012), Springer Berlin Heidelberg

[Aalst 16] Aalst, van der W.: *Process Mining*, Springer (2016)

[Aggarwal 15] Aggarwal, C.: *Data Mining*, Springer (2015)

[Brill 00] Brill, E. and Moore, R.: An Improved Error Model for Noisy Channel Spelling Correction, in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pp. 286–293, USA (2000), Association for Computational Linguistics

[Denisov 18] Denisov, V., Fahland, D., and Aalst, van der W.: Unbiased, Fine-Grained Description of Processes Performance from Event Data, in *Business Process Management*, pp. 139–157, Springer (2018)

[Dumas 18] Dumas, M., La Rosa, M., Mendling, J., and Reijers, H. A.: *Fundamentals of Business Process Management*, Springer (2018)

[Hompes 17] Hompes, B., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J., and Aalst, van der W.: Discovering Causal Factors Explaining Business Process Performance Variation, in *Advanced Information Systems Engineering*, pp. 177–192, Springer (2017)

[Senderovich 15] Senderovich, A., Weidlich, M., Gal, A., and Mandelbaum, A.: Queue mining for delay prediction in multi-class service processes, *Information Systems*, Vol. 53, pp. 278–295 (2015)

[Song 08] Song, M. and van der Aalst, W.: Towards comprehensive support for organizational mining, *Decision Support Systems*, Vol. 46, No. 1, pp. 300–317 (2008)

[Verenich 17] Verenich, I., Nguyen, H., La Rosa, M., and Dumas, M.: White-Box Prediction of Process Performance Indicators via Flow Analysis, in *Proceedings of the 2017 International Conference on Software and System Process*, pp. 85–94, Association for Computing Machinery (2017)

[Weijters 11] Weijters, A. and Ribeiro, J.: Flexible Heuristics Miner (FHM), in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp. 310–317 (2011)