

ビジネスプロセスにおける複数属性類似パス検索を用いた 将来予測

石田愛^{†1} 工藤道治^{†1}

様々なビジネスを遂行するためのシステムは常にログを生成している。これらのログを有効活用し、業務の現状把握や最適化を行うことが近年注目されている。本稿ではシステムログから生成された大量のビジネスプロセスに対して、複数の属性に注目しながら類似するパスを検索することによって現行のプロセスがどのような結果になる傾向があるのかを予測する手法を提案する。この手法を用いて過去のログから現状業務の取るべき次ステップの意思決定サポートが可能になる。

Prediction Using Similarity Search for Multiple Attributes on Business Processes

AI ISHIDA^{†1} MICHIHARU KUDOH^{†1}

The information system which is necessary for company to continue with business as usual generates a massive amount of logs. Business Process Analytics, techniques that understand an actual workflow and optimize it using based on theses system logs have attracted attention in recent years. In this paper, we propose a similarity search technique to predict a tendency of process outcome utilizing multiple attributes of event logs. The technique supports business decision by indicating a trend of a current process.

1. はじめに

昨今、膨大な量のデータを解析して新しい知見を見出すとする技術が注目を集めている。例えば、Twitterやフェイスブックのようなソーシャル・ネットワーク・サービス(SNS)の入力内容を分析し市場における製品の評判を分析したり、地域によるインフルエンザ流行度合いを推定したりする試みが知られている。そしてそれは企業や組織が持つ様々なデータに関しても例外ではない。業務を遂行する際に使用するシステムは大量のイベントデータをログとして記憶しており、それらのデータを利用することによって現行業務の可視化や効率化のための分析が可能になる。本稿ではこれら業務分析の中でも、ビジネスプロセスという業務の小さなタスクのつながりを表したもの(図1)に対する分析、ビジネスプロセス分析について扱っていく。

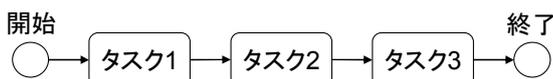


図1 ビジネスプロセスのイメージ

1.1 ビジネスプロセス分析

ビジネスプロセス管理ツールの登場によりビジネスプロセ

スは、GUIによる設計、シミュレーションによる性能予測、実装の生成、サービスの配置、モニタリング、パフォーマンス分析、業務改善案の提示などが効率よく行えるようになった。例えば、IBM Business Process Manager [1] は上記の機能をすべて備えている。IBM Blueworks Live [2] はシンプルなウェブアプリケーションを用いてすぐに業務プロセスの設計を行うことができる。また Prom [3] という業務分析用フリーソフトも提供されている。

一方で、これらの管理ツールを用いずに実装、運用されたビジネスプロセスでは、このような恩恵を得ることはできない。しかし様々な事情により、実際にはこのようなプロセスは多く存在する。そうしたプロセスでは、実際に運用されている業務フローが把握できず、問題点の発見が困難になる。そのような場合は、業務を遂行する際に用いたシステムから得られるデータを用いてビジネスプロセスを生成する。

情報システム上でビジネスプロセスを実現するとき、各システムにおいては、イベントという分割された単位の処理が行われる。システムがイベントを処理したとき、そのイベントに関する情報がログに書き込まれる。このようなログをイベントログと呼ぶ。多くの場合、イベントログは故障時の問題判別や、ビジネス上の事故の証拠として使われている。

ビジネスプロセス分析に用いることができるイベント

^{†1} 日本アイ・ビー・エム株式会社
IBM Japan

ログは、少なくともケース ID、タスク名の 2 つが記録されている必要がある。表 1 に、イベントログの例を示した。この例では、1 つのイベントの情報を 1 行に格納している。ケースとは複数のイベントの集合であり、一連のイベントが何かしらの意味を持つ流れを示している。例えば販売プロセスであれば各注文の受注・支払い確認・発送準備・発送といったイベントの一連の流れをケースと呼ぶ。そしてこれらケースを一意に特定できるものがケース ID である。このようなイベントログから再構築されたワークフローと呼ばれる有向グラフがビジネスプロセスとなる (図 2)。

表 1 イベントログの例

時刻	ケース ID	タスク名	作業員	...
t1	1	Task1	中村	
t2	1	Task2	篠崎	
t3	1	Task3	和田	
t4	2	Task1	山村	
t5	2	Task4	小野	
t6	2	Task5	山村	
t7	2	Task3	渡辺	

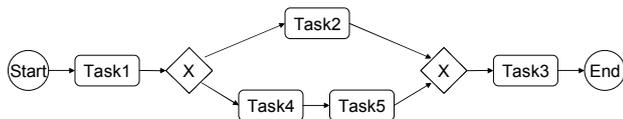


図 2 Process Discovery によって表 1 のイベントログから再構成したワークフロー図

こうして生成されたビジネスプロセスに対する分析手法として様々なものが提案されているが、それら分析はおおよそ以下の 5 つのコアとなる技術に分類できる。

- **ディスカバリー:** ログの列からビジネスのフローを復元し、グラフを作成する。巨大なビジネスでは、追加と修正を繰り返した結果その詳細なビジネスフローが誰もわからなくなってしまうケースが多々存在する。このような状況からビジネスを再び理解できるようにし、コントロールを取り戻すための手助けとなる。
- **統計:** ログの列から、ビジネスの統計情報を集計し、経営者に分かりやすい形で提示する。時間がかかっているプロセス、処理が集中しているプロセス、時間帯や担当者によるサービス品質の違いなど、ビジネスのボトルネック・改善箇所を発見する。
- **検証:** 正常なパスを辿って終了したケースと、そうでないケースに分類する。ビジネスにおいて例外処理は

必要不可欠だが、処理コストが高く、またプロセスフローを複雑にする要因ともなる。検証は、こうした例外処理を把握し、正常処理に対して一定の割合以下に抑える事で、従業員の負荷を軽減させる。

- **因子分析:** コスト超過など異常が発生したケースに対して、どのプロセスが原因となったのかを特定する。現実のビジネスでは、一つのケースを処理するために多くのプロセスを経るため、処理時間が伸びた場合にどのプロセスを改善すれば良いのかといった判別が困難となる。
- **予測:** 現在手に入るイベントログの情報から、処理途中のプロセスがその後どのように処理されるのかを予測したり、プロセスを改善した場合にどのように変化するかを予測したりする。

本稿ではこれらコア技術の中でも予測技術として、複数のイベント属性を基にプロセスを構築し、実行中のプロセスとの類似度を求める手法を提案する。より正確に現在のプロセスと類似している過去のプロセスを検索し、検索結果の統計情報から現在のプロセスの今後を予測・提案することが可能となる。

1.2 プロセスの将来予測

ある時点で実行中のプロセスがその後どのような処理を辿り、どのような結果になる傾向があるのか予測するには、過去に実行された処理のイベントログから実行中のプロセスと同様のプロセスを発見し、それらがどのような結果になったのかを見ればよい。過去ログから生成された膨大な量のプロセスの中から、現在実行中のプロセスと似たプロセスを検索するためにはグラフの類似度を求める必要がある。プロセスグラフの類似度を求めるには、2 つのグラフがどのくらい違うのかを示すグラフ編集距離[4]を求める手法がある。この手法では注目するイベント属性、即ちワークフローを生成する際に利用したイベントログのとある属性についての類似度を求めることは可能だが、それ以外のイベントログが持つ属性については考慮されていない。例えば、図 2 のワークフローは表 1 のタスク名という属性の値を用いて生成されているため、このワークフローの類似度を求めるとタスク名の推移が類似しているプロセスを発見することができる。しかしながら、イベントログのタスク名だけでなく作業員という属性の値の推移についても類似しているプロセスを見つけたいと考えた場合、上記の方法では検索が困難になる。

そこで本提案手法では、ユーザが指定したイベントログ内の属性 (複数選択可) 毎にプロセスグラフを生成し、各類似度を総合して検索する方法を用いている。これにより複数属性を加味した類似検索が可能になり、ゆえに、よりユーザが必要とする情報を含んだプロセスの将来予測が可

能になると考えられる。

本稿では、第2章にていくつかの関連研究について紹介した後、第3章でより詳細に提案手法を説明する。さらに第4章においてそれらの手法を用いた実験とその結果について記述し、第5章にて考察を行う。

2. 関連研究

過去のデータから現在のプロセスがとるべき次ステップを提案する手法は、レコメンデーション技術として[6][7][8]などが提案されている。[6]は過去のプロセスの実行データから現在のプロセスの次ステップを提案するが、過去のトレースと現在のトレースを比較する際に前処理として抽象化を行い、過剰適合を防いでいる。[7]は過去のデータからユーザ毎にプロセスモデルを学習し、それを基に次ステップの提案を行う。これら2手法はどちらも過去データのプロセスの構造にだけ注目している。一方[8]はプロセスモデルを構築し、さらにそのモデルにプロセスのメタデータから生成したアノテーションを加えることによりレコメンデーションの精度を上げている。しかしこれらの手法では、過去データのモデルを構築する際にイベントが持つ多くの属性の内一属性のフローにしか注目しておらず、その他の属性フローは活用されていない。本提案手法では、これらの捨てられていた属性情報も活用し将来の予測を行う。

3. 類似検索による将来予測

前述したように、ある時点で実行中のプロセスがその後

どのような処理を辿り、どのような結果になる傾向があるのか予測するには、過去に実行された処理のイベントログから実行中のプロセスと同様のプロセスを発見し、それらがどのような結果になったのかを見ればよい。図3は本研究で提案する、実行途中のプロセスのその後を予測するためのシステムの流れを示している。

予測のためのシステムでは、分析対象となる特定の業務に関するデータがイベントログとして保存されていることを前提とする。それらのデータを入力とし、前処理のフェーズにて過去に収集されたイベントログとそのログ内の注目すべき属性を基に、有向グラフであるプロセスフローを生成する。生成されるプロセスフローの数は注目する属性の数に比例する。

ランタイムのフェーズではユーザが予測を必要とする時点までの実行中のプロセスについて、前処理と同様に注目する属性に基づいてプロセスフローを生成する。前処理において生成したプロセスフロー群の中から実行中のプロセスと類似しているものを抽出するためには、実行中のプロセスが停止している時点までのプロセスフローをサブグラフとして自身のプロセス内に保持しているプロセスを探す必要がある。この処理を注目すべき属性ごとに行い、算出されたグラフ編集距離にて求められた類似度を足し合わせたものを2プロセスフロー間での類似度とする。この類似度がある閾値を越えたプロセスを類似プロセスとする。類似プロセスとしてされたプロセスフロー群のアウトカムの統計値を示すことによって、過去の傾向から実行中のプロセスがどのようなアウトカムを出力する傾向があるのか示すことが可能となる。

以降本章では各処理の詳細な手法について示していく。

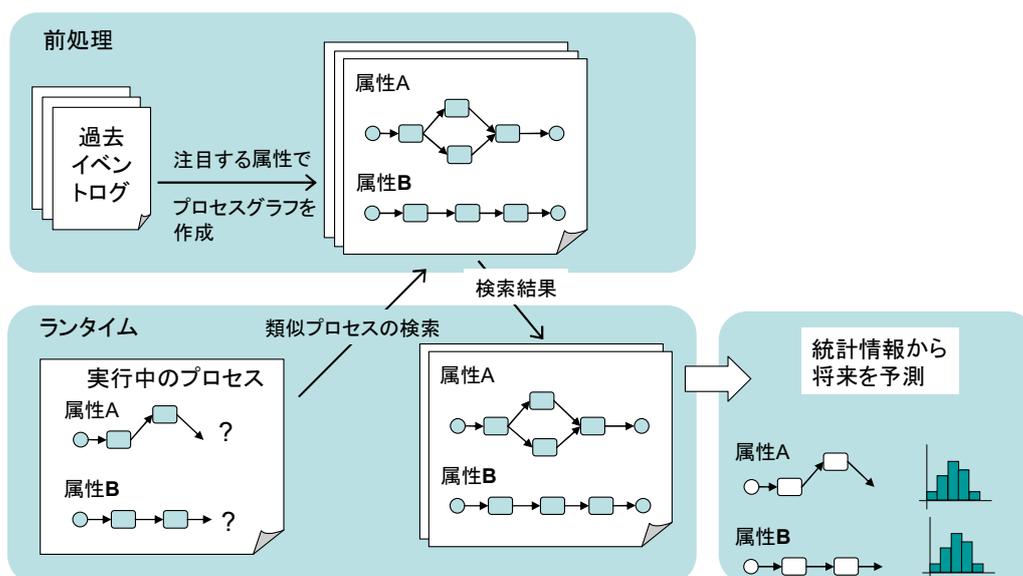


図3 提案システムの処理の流れ

3.1 プロセスフローの生成

イベントログからのプロセスフロー生成には, Aalst らの α アルゴリズム [5] を利用する. α アルゴリズムとは, イベントログからワークフローを再構築する手法の 1 つである. α アルゴリズムではまずイベントログに出現する 2 イベント間の順序関係を列挙する. $a \succ b$ をイベント a とイベント b が連続して出現しており, かつ a の前に b が出現しているとする, 2 イベント間の順序関係は, 以下の 3 種のいずれかとなる.

- $a \rightarrow b \iff a \succ b \wedge b \not\succeq a$
- $a \# b \iff a \not\succeq b \wedge b \not\succeq a$
- $a \parallel b \iff a \succ b \wedge b \succ a$

このようにしてイベントログの同ケース内のイベント間の順序関係について列挙した後, 列挙された組み合わせを集約しその順序関係を満たすようにグラフを再構築していくとワークフローが生成される.

本提案手法ではユーザが指定した属性すべてに対してワークフローを生成する必要がある. 例えば, 表 1 のようなイベントログに対して, ユーザがタスク名の流れと作業者の流れの両方に注目したプロセスの類似度を求めたいと考えたとする. この場合, ひとつのプロセスデータに関して 2 つのワークフローが生成される. ひとつはタスク名をノードとした有向グラフであり, もうひとつは作業者をノードとした有向グラフ (図 4) である.

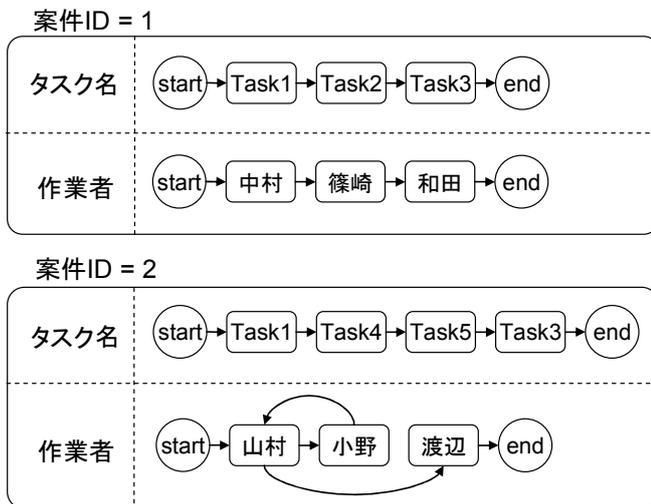


図 4 イベントログ (表 1) の属性 “タスク名” と “作業 者” から生成されたワークフロー

3.2 比較対象のサブグラフ検出

過去のイベントログから生成されたプロセスフローがある業務の全体の流れを示しているのに対し, 実行中のプロセスから生成されたフローは業務の一部しか示していない.

そのため, そのまま 2 グラフ間の類似度を求めようとしても, 正しい結果が算出されない. そこで過去のイベントログから生成されたフローの中から, 実行中のプロセスと同等と思われる部分を抽出し比較する (図 5).

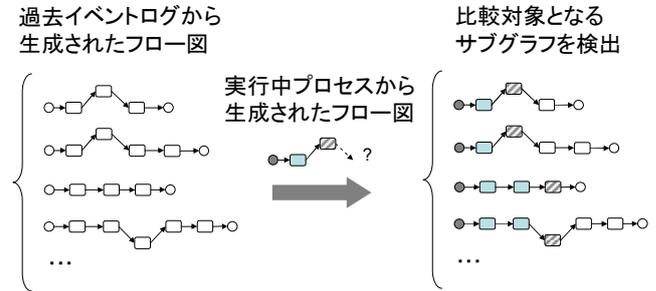


図 5 実行中のプロセスと過去のプロセスを比較するため, 比較対象とするサブグラフを検出する

比較のためのサブグラフ検出で一番簡単な方法は, 実行中プロセスの最初のノードと最後のノードと同等のノードを過去のイベントログから生成されたフローから見つけ出し, その 2 ノード間のフローを比較対象のサブグラフとする手法である. ただし, 該当するノードが見つからないからといって, 比較対象のサブグラフが存在しないと判断してしまうと類似検索の幅を狭めてしまうため注意が必要である. 該当ノードが見つからなかった場合はそのノードと比較的近いと思われるノードを, ラベルマッチングの技術やメタデータを利用し検出する [9] 必要がある.

3.3 グラフ編集距離

前節で検出されたサブグラフと実行中のプロセスフローの類似度を計算するために, 本研究ではグラフ編集距離 [10] [11] を利用した. グラフ編集距離の基本的な考え方は, あるグラフを他のグラフに変換するのに必要な最小の編集数を 2 グラフ間の相違とすることである. 編集操作は, ノードとエッジの挿入, 削除, 置換の 3 種で構成され, 編集操作の数 e_d で編集距離が定義される. 全てのグラフの組 G_1 と G_2 において, 特定の編集操作で G_1 を G_2 に変換する編集経路の集合 $h(G_1; G_2) = (e_{d1}, \dots, e_{dk})$ (ここで各 e_{di} が編集操作を表す) が存在する. 一般的に, 2 つのグラフ間の編集経路は複数存在する. どの編集経路が最良かを評価するために, 編集コストを導入する. 基本的な考え方は, それぞれの編集操作によって起こるひずみを, 編集コスト c で表すことである. 2 つのグラフ G_1 と G_2 の間の編集距離 d は, $d(G_1; G_2)$ で表され, 1 つのグラフを他のグラフに変換する最小コストである編集経路のコストとなる. それは以下の式によって表される.

$$d(G_1, G_2) = \min_{(e_{d1}, \dots, e_{dk}) \in h(G_1, G_2)} \sum_{i=1}^k c(e_{di})$$

ここで、 $h(G_1;G_2)$ は G_1 から G_2 へ変換する編集経路のセット、 $c(e_d)$ は編集操作 e_d のコストを示す。 $d(G_1;G_2)$ が小さければ小さいほど2グラフ間の距離が小さい、即ち類似度が高いということができる。

3.4 総合的な類似度

本提案手法では注目すべき属性毎にプロセスフローが生成されるため、1組のプロセスに対して複数の類似度が算出される。そのためそれら類似度の和をもって総合的な類似度とする。選択された属性数が多い場合や、選択した属性の中でも重要度が異なる場合は各属性の類似度に重み c を掛けた値の和を用いる必要がある。

$$d(G_n, G_r) = \sum c_{attr} d(G_{nAttr}, G_{rAttr})$$

3.1 節にて表 1 のイベントログから生成したプロセスフロー (図 4) を例に簡単に説明する。案件 ID = 1 のタスク名をノードとするプロセスフローを G_{1t} , 作業者をノードとするプロセスフローを G_{1p} , 案件 ID = 2 のタスク名をノードとするプロセスフローを G_{2t} , 作業者をノードとするプロセスフローを G_{2p} とする。今実行中のタスク名をノードとするプロセスフロー: G_{rt} , 作業者をノードとするプロセスフロー: G_{rp} を図 6 のように定義する。

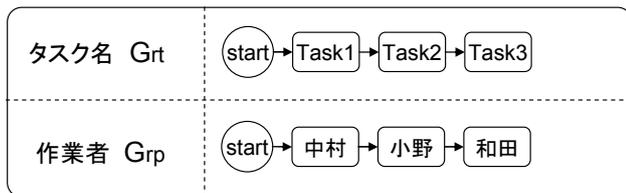


図 6 実行中のプロセスの定義

この場合各類似度は以下の通りとなる。

$$d(G_1;G_r) = d(G_{1t};G_{rt}) + d(G_{1p};G_{rp}) = 0 + 1 = 1$$

$$d(G_2;G_r) = d(G_{2t};G_{rt}) + d(G_{2p};G_{rp}) = 2 + 4 = 6$$

この結果より案件 ID = 1 の方が実行中のプロセスと類似していることが分かる。実際のシステムでは閾値を設定し、その閾値を下回ったものを類似プロセスとする。

3.5 統計データによる将来予測

前述の手法により実行中のプロセスと類似するプロセスを選択した後は、それらプロセスの処理時間やコストといったアウトカムの統計値をユーザに示す。過去の同じようなプロセスがどのような結果になったのかを示すことによって、実行中のプロセスが今後どのような展開になるのか予測可能となる。さらに類似検索結果のプロセスの集団がある指標でクラスタリングすることによって、次に行うべきタスクや選択すべき担当者など、未来の行動が結果にどのような影響を及ぼすのか、判断することが可能となる。

4. 評価実験

本節では 3 節で説明した提案手法についてテストデータを用いて実行し、その有用性について評価する。

4.1 評価データ

本提案手法は業務システムのログを扱うことを想定している。しかしながら実際の業務データを実験にて使用するのは難しいため、我々はまず業務をシミュレーションするシステムを実装した。業務シミュレーターではランダムで業務イベントが発生し業務プロセスを進めていく。業務のイベントの種類は 45 種類でそれらがすごろくのように、ある程度の発生順序の規則性を持った状態でランダムに発生する。特定のイベントが発生するとそれ以降イベントが発生しなくなるため、それまでの一連のイベントをケースとして、イベントログに保存していく。このようにして生成された 3 万ケース分のイベントログが実験で使用されるデータとなる。表 2 は実際に生成されたイベントログの一部である。イベントが発生した時刻、プロセスの ID、イベントの名前の他に、イベントの担当者の ID、プロセスの状況を表すステータス、作業コストが各イベントログの属性として保存されている。

表 2 生成されたイベントログの一部

時刻	ケース ID	イベント名	担当者	ステータス	コスト
t1	1	Task1	p1	s1	1000
t2	1	Task2	p10	s2	2000
t3	1	Task3	p3	s2	3000
t4	2	Task1	p4	s1	1000
t5	2	Task4	p8	s3	2500
t6	2	Task5	p4	s3	1500
t7	2	Task3	p5	s2	3000

従来手法ではイベント名だけに注目してプロセスを生成、類似度を計算するしかなかったが、本提案手法では複数の属性を考慮した類似度分析が可能のため、今回の実験ではイベント名に加え、業務のステータスも注目すべき属性として扱う。イベントログ内に出現するイベント名は 45 種類、業務のステータスは 8 種類である。プロセスのアウトカムは一プロセスにおける作業コストの和とし、それが少ないほうが効率のよい業務として扱う。

これらのデータに対し、類似度検索をする実行中のデータは同様の業務シミュレーターにて生成する。ただし、イベントログ生成時とは異なり終了イベントに行く前にシミュレーションを止め、実行中のプロセスを表現する。

4.2 実装・評価

本提案手法を実現するプロトタイプを実装した。また実

験では、従来の1属性だけに注目したグラフ編集距離を用いて類似プロセスを抽出する手法も試行し、本提案手法との比較を行う。

4.2.1 類似プロセス検索

グラフを類似と定義するための適切な閾値 θ は、適用するデータに依存するためユーザに指定させることも可能だが、本実験で用いたデータに対しては、実験から得られた同一属性における値の種類 n_{attr} を用いて $\theta = n_{attr} * 0.05$ としている。この値以下の編集距離を持つ2グラフを類似とする。図7、表3は従来手法である1属性の編集距離を用いた際に検索できた類似プロセスの件数と、提案手法である複数属性（今回は2属性）の編集距離を用いた際に検索できた類似プロセスの件数を比較したものである。検索条件として使用した実行中のプロセスは20件で、図7では類似プロセス件数の少ない順に並べてある。

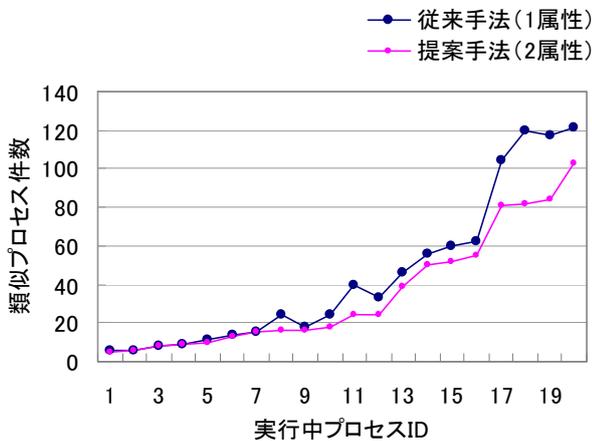


図7 従来手法と提案手法による類似プロセス検索結果件数の違い

表3 従来手法と提案手法による類似プロセス検索結果件数の違い

実行中プロセスID	1	2	3	4	5	6	7	8
従来手法の検索件数	6	6	8	9	11	14	15	24
提案手法の検索件数	5	6	8	9	10	13	15	16

9	10	11	12	13	14	15	16	17	18	19	20
18	24	40	33	46	56	60	62	104	120	117	121
16	18	24	24	39	50	52	55	81	82	84	103

図7、表3から明らかなように、提案手法の検索結果件数の方が従来手法の件数よりも常に小さくなっている。その傾向は特に類似プロセスが多い場合に顕著である。実際に検索されたプロセスを観察すると、提案手法の方がユーザ

が求めている類似プロセスを取得できていることが確認できる。つまり提案手法はより効果的に、ユーザが必要とする情報だけを持つ類似プロセスの取得が可能で、そのため従来手法よりも検索結果の件数が少なくなっていると言える。

4.2.2 統計情報を用いた将来予測

類似プロセス検索の結果得られたプロセスの集合に対して統計処理を適用することで、現在のプロセスと類似したプロセスが今後どのような結果になるのか傾向を示すことが可能となる。また、プロセス集合をある特定イベントを基に複数に分割し統計情報を示すことで、そのイベントを選択するかどうかでどう結果が変化するかを予測することもできる。図8は類似検索の結果のプロセスが最終的にどのくらいコストがかかったのかを表すヒストグラムである。類似検索の結果、実行中のプロセスと同じようなサブグラフを持つプロセスが複数選択されるが、さらにそのプロセス内であるイベント（イベントAと仮定）を持つプロセスとそうでないプロセスに分割するとコストに関する2種類のヒストグラムが生成される。これは、実行中のプロセスがその後イベントAを実行した場合とそうでない場合でどのようなアウトカム（コスト）の差が出るのかについて表したグラフとなる。今回の結果ではイベントAを実行したほうが全体的にアウトカムであるコストが高くなる傾向があることが分かる。このような情報をユーザに示すことで今後ユーザが実行中のプロセスをどのように進めていけばよいのかという意思決定を助けることが可能となる。

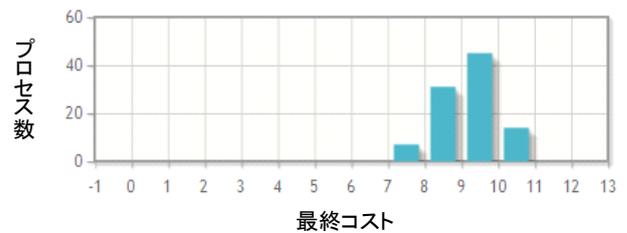


図8(a) イベントAを持つプロセスの最終的な総コストを表すヒストグラム

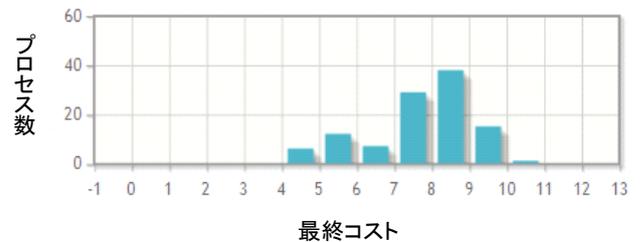


図8(b) イベントAを持たないプロセスの最終的な総コストを表すヒストグラム

5. まとめ

本稿では、システムログから生成された大量のビジネスプロセスに対して、複数の属性に注目しながら類似するパスを検索することによって現行のプロセスがどのような結果になる傾向があるのかを予測する手法を提案した。さらに実験で従来手法と検索の結果を比べ、本提案手法が従来手法よりも効果的に類似プロセスを取得できることを示した。この手法を用いることによって過去のログから現状業務の取るべき次ステップの提案が可能になる。

本提案手法はランタイムで実行することを前提としているため、今後は実行時間も考慮に入れながらのアルゴリズムの改良が求められる。また本提案手法の適応先として、ビジネスプロセスだけでなく医療プロセスや他のプロセス化できるデータに対しても対応できるように、汎用性を高めていく予定である。

参考文献

- 1) IBM: “IBM Business Process Manager”, <http://www-06.ibm.com/software/jp/websphere/bpm/>
- 2) IBM: “IBM Blueworks Live”, <http://www-06.ibm.com/software/jp/websphere/bpm/bpm-blueprint/>
- 3) “Prom”, <http://www-06.ibm.com/software/jp/data/ecm/bpm.html>
- 4) H. Bunke and G. Allerman, Inexact graph matching for structural pattern recognition, " Pattern Recognition Letters, Vol. 1, pp. 245-253 (1983)
- 5) W.M.P. van der Aalst, A.J.M.M. Weijters, and L. Maruster, Workflow Mining: Discovering process models from event logs, *IEEE Transactions on Knowledge and Data Engineering*, vol 16 (2003)
- 6) Schonenberg, H., Weber, B., van Dongen, B.F., van der Aalst, W.: Supporting Flexible Processes through Recommendations Based on History. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 51–66. Springer, Heidelberg (2008)
- 7) Dorn, C., Burkhart, T., Werth, D., Dustdar, S.: Self-adjusting recommendations for people-driven ad-hoc processes. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 327–342. Springer, Heidelberg (2010)
- 8) Hamid Reza Motahari-Nezha, Claudio Bartolini, Next Best Step and Expert Recommendation for Collaborative Processes in IT Service Management, BPM'11 Proceedings of the 9th international conference on Business process management pp50-61 (2011)
- 9) R. M. Dijkman, M. Dumas, B. F. van Dongen, R. Karik, and J. Mendling. Similarity of business process models: Metrics and evaluation. Working Paper 269, BETA Research School, Eindhoven, The Netherlands (2009)
- 10) D. Conte, P. Foggia, C. Sansone and M. Vento, Thirty years of graph matching in pattern recognition, " International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 265-298 (2004)
- 11) H. Bunke and G. Allerman, Inexact graph matching for structural pattern recognition, " Pattern Recognition Letters, Vol. 1, pp. 245-253 (1983)
- 12) A. Sanfeliu, and K. Fu, A distance measure between attributed relational graphs for pattern recognition, " IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13, No. 3, pp. 353-362 (1983)