

[解説]

プロセスマイニング・サーベイ (第02回: ツール)

飯島 正[†], 田端 啓一[‡], 斎藤 忍[‡]

1 はじめに

プロセスマイニング・サーベイの第2回は、ツール編である。プロセスマイニングの入力データに関しては第3回で、各種マイニングアルゴリズムの詳細は第4回で解説するが、詳細に入る前に、まずツールに触れていただくことで読者にイメージをつかんでいただくことが目的である。第7節「おわりに」にも述べるが、何よりも第3節で解説する ProM をインストールし¹、代表的なマイニングアルゴリズムをつかい例題データで制御フロー発見を試みていただくことをお薦めする。

前回(サーベイ第1回)で解説したようにプロセスマイニングの核をなす制御フロー発見は、1990年代後半に研究開発が開始され²、2000年代前半には複数の種類のプロセスマイニングのツールが構築されていた。たとえば、構造化モデルを前提とする Guido Schimm の Process Miner (2002年)³ [1]、確率的アクティビティグラフ (stochastic activity graph) を対象とする Herbst と Karagiannis の InWoLvE(2003年)^[2]、 α -アルゴリズムとその拡張を実装した van der Aalst らのグループによる MiMo(2002年)^[3, 4]、EMiT(2002年)^[5, 6, 7]、

Little Thumb(2003年)^[8] などがある⁴。これらの初期のツールの比較は、文献^[9]に詳しい。

また、同時期に開始された VIP (Verification of Information systems by evaluating partially-ordered Petri net runs) プロジェクト⁵で開発され、現在までメンテナンスが続けられている VIP Tool [10, 11, 12, 13, 14, 15] もある。VipTool は、実行トレースのようなシーケンスではなく、半順序構造をもった実行経路 (partial-ordered run) の集合からそれを満たすペトリネットを合成する。

そのほか、命令的 (imperative) にタスクの順序を規定するのではなく、宣言的 (declarative) に時制論理に基づいたタスク間の制約関係の集合を与える ConDec モデリング言語 (2006年)^[16]⁶に始まる宣言的プロセスモデリングツール Declare システム⁷ [17, 18, 19] もあり、ProM のプラグインも Declare Package にて導入可能である⁸ [20]。

2008年以降には、状態ベースモデルによる領域 (Region) 理論に基づくアルゴリズムを実装した Genet⁹[21] や Rbminer¹⁰[22] がある。

本稿では、これらの個別のツールについての調査報告も行うが、オープンソースのプロセスマイニングツールとしては、いろいろなマイニングアルゴリズムやデータ変換ツールが ProM¹¹ [23, 24, 25, 26] (2004年以降) に集約されているので、特にこれの試用をお薦めする。ProM は単独のアルゴリズムを実装したツールではなく、プロセスマイニングに関わる共通要素を実装した一種のフレームワークないしワークベンチ (実験台) であり、いろいろなプロセスマイニングをアルゴリズムをプラグインとして実装することで、そのアルゴリズムを動かすことができるプラットフォームとして機能する。

A Survey on Business Process Mining
— 02: Tools —

Tadashi Iijima[†], Keiichi Tabata[‡], and Shinobu Saito[‡]

[†]Faculty of Science and Technology, Keio University

[‡]Nippon Telegraph and Telephone Corporation

[†]慶應義塾大学・理工学部

[‡]日本電信電話株式会社

[解説] 2016年5月24日受付。

©一般社団法人 情報システム学会

¹ 2016年6月10日に最新版である「研究者向け ProM 6.6」と「エンドユーザー向け ProM Lite 1.1」が、リリースされているが、本稿では執筆時点の最新版である 6.5.1 (と一部 ProM 5.2) を取り上げている。しかし、インストールや基本的な利用に関しては大きな相違はない。

² 前回は述べたように、有限状態マシン (FSM) や隠れマルコフモデルの学習を含めるとさらに遡ることができるがそれは別物と考える

³ Guido Schimm のツールとしては、<http://www.processmining.de/> から Process-Mining-Workbench と Process Modeler がダウンロードできるが、残念ながら十分なドキュメントが見あたらない。

⁴ 4年号は、確認できた各ツールの関連文献の発表年であり、必ずしも開発時期とは一致していない

⁵ https://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml

⁶ 後に Declare 言語と名前が変更された

⁷ <http://www.win.tue.nl/declare/>

⁸ <http://www.win.tue.nl/declare/declare-miner/>

⁹ <http://www.cs.upc.edu/~jcarmona/genet.html>

¹⁰ <http://www.cs.upc.edu/~jcarmona/rbminer/rbminer.html>

¹¹ <http://www.processmining.org/prom/start>

実際のところ、MiMoやLittle Thumb等で実装された α -アルゴリズムやその拡張(たとえばヒューリスティックマイナー)などもProM上に実装されている。2004年にリリースされたProM 1.1では、29種のプラグインが取り入れられており、その内訳は、マイニングプラグイン(6種)、解析プラグイン(7種)、インポートプラグイン(4種)、エクスポートプラグイン(9種)、変換プラグイン(3種)、フィルタプラグイン(0種)である。その後、2006年にリリースされたProM4.0では142種のプラグイン、2009年にリリースされたProM5.2では274種(2009年9月時点)のプラグインが含まれている。その後、プラグイン数は、2012年にリリースされたProM6.2では、2012年8月時点で68のパッケージに分散された468種だったものが、2013年5月には676種(99パッケージ)と増えている¹²。プロセスマイニングとそのツールに関するアンケート調査では、特にProMに関する設問も設けており、興味深い^[27]。

プロセスマイニングのツールの歴史的な展望(特に2004年~2013年5月までのProM)としては、2013年5月28日に開催されたProcess Mining Camp 2013におけるvan der Aalstの基調講演^[28]に詳しい。

さらに本解説では、商用ツールとしてFluxicon社のDisco¹³^[29]と富士通のInterstage Business Process Manager¹⁴も取り上げる。ProMは多くのアルゴリズムを共通のフレームワークで実行することができ、解析ツールや、各種の入力データのインポートツールや、各種の出力データのエクスポートツールが統合されているワークベンチ(実験台)という面もあり、現状では、その分、どうしてもツールとしての安定性は、商用ツールにかなわない面もあるようである。

プロセスマイニング関連のツールとしては、パースペクティブとして、制御フロー、データ/ルール、組織といった分類があるが、今回は制御フロー発見のツールに焦点をあてて取り上げ、データに関するツールは次回(第3回)「データ編」で触れることとする。組織パースペクティブのツールとしては、人間関係の社会ネットワークを発見するMiSoNなどがあり、ProMにも、そのための

¹²各パッケージのドキュメントは、<http://www.promptools.org/doku.php?id=packdocs>にある

¹³<http://www.fluxicon.com/disco/>

¹⁴<http://www.fujitsu.com/jp/products/software/middleware/business-middleware/interstage/products/bpmgr/>

Social network minerプラグインが実装されている。しかし、本サーベイでは、そうした組織パースペクティブに関しては範囲外として取り上げないこととする。

また、Woflan¹⁵等の一般的なペトリネットの性質(構造正当性、性能など)の解析ツールだけでなく、Conformance CheckerやLTL Checkerなどの適合性解析ツールについても今回は取り上げない。

2 ツールやアルゴリズムのテスト用データ

今回のツール編は、早い段階から実際にツールを使ってイメージをつかんでいただくことを目的としている。そのため、個々のツールで実装されているアルゴリズムやデータの詳細に立ち入る前に、主に実際に入手し試用することが可能なツールに関して解説している(データの詳細は次回、アルゴリズムの詳細は次々回に取り上げる予定である)。

だが、ツールを試すためには、入力となるワークフローログが必要である。実適用の折には、実際の情報システムが運用中に出力した生のログデータから、解析目的に必要なデータを抽出しデータ形式を加工して作成することになる。もっとも、例えば監査目的に蓄積保管されてきたログデータが、マイニング(制御フロー発見)に十分に活用できるとは限らない。目的によっては、その目的に応じた必要な属性データを従来から記録しているイベントデータに加えて出力するサブモジュールを既存の情報システムに後付けすることも必要となるかもしれない。

しかし、ツールやそこで実装されているアルゴリズムを試用することが目的であれば、特に解析対象の情報システムを持たず、テストに使用するログデータが手元にないということもありうる。その場合のテストデータの入手方法には、下記のようなものがある:

- (1) ツールの配布パッケージに同梱されている例題データ、
- (2) ツールの配布サイト、書籍のサポートサイト、チュートリアルサイト等のWWWページ、

¹⁵<http://www.win.tue.nl/woflan/doku.php>

- (3) ログデータのリポジトリ,
- (4) プロセスモデルのシミュレータによるログ生成.

但し、データファイルのデータ形式には、テキストファイル、CSV形式、MXML (Mining eXtensible Markup Language) 形式¹⁶[30, 31], XES (eXtensible Event Stream) 形式¹⁷[32] などがありうる。データ内容も、ケース(事案)IDとイベント列以外に、タイムスタンプ、頻度、関与者、資源などのイベントに関する各種属性データも伴うものがあるので利用する前には内容を確認してほしい。試用するツール毎に、適したデータ形式のものを選択するか、必要な形式へ加工ないし変換して使用する必要がある。データ形式の詳細に関しては、次回(第3回: データ編)を参照してほしい。

2.1 ツールの配布パッケージに同梱されている例題データ

ツールの配布パッケージに同梱されている例題データは、基本的に、そのツールで利用できることが保証されているので、気軽に試すことができる。例えば、後述するEMiTの配布パッケージにはInConcert, Pnet+, StaffWareといった形式のログデータが例題として同梱されており、EMiTは、これらのログをXML形式に変換して読み込むことができる。

2.2 ツールの配布サイト、書籍のサポートサイト、チュートリアルサイト等のWWWページ

例えば、文献^[33]で使われているログデータは、http://www.processmining.org/event_logs_and_models_used_in_bookで配布されている¹⁸。

また、後述するツールProMのチュートリアル・サイト¹⁹でも例題(Exercise)データを配布している²⁰。本稿でも、この例題データを第3節と

¹⁶<http://www.processmining.org/logs/mxml>

¹⁷<http://www.processmining.org/logs/xes,http://www.xes-standard.org/>

¹⁸—括配布ファイル: http://www.processmining.org/_media/processminingbook/event-logs-process-mining-book.zip

¹⁹<http://www.processmining.org/prom/tutorials>

²⁰<http://www.promtools.org/prom6/downloads/example-logs.zip>

第5.2節でも利用する。

2.3 ログデータのリポジトリ

ログデータのリポジトリに、3TU.Datacentrumサイト²¹がある。このサイトのリポジトリの、IEEE TF on Process Mining - Event Logsカテゴリ²²の下に、本稿執筆時点では、実生活イベントログ(Real life Event Logs)²³サブカテゴリと、合成イベントログ(Synthetic Event Logs)²⁴サブカテゴリ、および、製造プロセス処理ログデータ(Production Analysis with Process Mining Technology)がある。すなわち、論文等で実験や比較に使われている人工データと、実適用にむけて現実のデータの難しさを知るために有用な実データが含まれている。ここでは、表1に示すような実生活イベントログ(Real life Event Logs)サブカテゴリ、表2に示すような合成イベントログ(Synthetic Event Logs)サブカテゴリ(tpn, pnmlはペトリネットモデルのデータ形式)と表3に示すようなログデータが登録されている。ここでは、各ログデータのタイトルと簡単な情報を列挙するにとどめるので、詳細は直接リポジトリを参照してほしい。

2.4 プロセスモデルのシミュレータによるログ生成

2.5 MiMo

詳細は後述するが、制御フロー発見ツールの一つであるMiMoツールは、シミュレーション機能を備えた階層的WF-ネットの図的エディタであるExSpectの上に構築されている。そのマイニングアルゴリズム自体が、実行可能階層的な階層的WF-ネットで実装されており、解析対象であるプロセスモデル(階層的WF-ネット)とExSpect上で結合して実行することで、解析対象のプロセスモデルをシミュレーション実行させて実行トレースを生成しながら、そのログをファイルに保存したり、マイニングアルゴリズムへ入力させること

²¹<http://data.3tu.nl/repository/>

²²http://data.3tu.nl/repository/collection:event_logs

²³https://data.3tu.nl/repository/collection:event_logs_real

²⁴https://data.3tu.nl/repository/collection:event_logs_synthetic

BPI Challenge 2011	実生活イベントログ - 病院ログ	XES	2011
BPI Challenge 2012	ローン申請プロセスのイベントログ	XES	2012
BPI Challenge 2013	Volvo IT のインシデントと問題管理のログ	XES	2013
BPI Challenge 2014	ICT 企業の ITIL サービス管理ツールのアクティビティログ	CSV	2014
BPI Challenge 2015	オランダの 5 自治体から提供されたデータ	XES	2015
BPI Challenge 2016	UVW(オランダの被用者給付制度機構) 提供のデータ	CSV	2016
CoSeLoG プロジェクト	環境的許可申請プロセス	XES	2014
CoSeLoG プロジェクト	環境的許可申請プロセスの受付フェーズ	XES	2014
道路交通管理プロセス		XES	2015
複数の個人の日常生活行動		XES	2015

表 1: 3TU.Datacentrum に登録されている, 実生活イベントログ (Real life Event Logs)

CPN Tools による合成イベントログ (review example - large.xes.gz)		XES	2010
CAiSE 2011 Forum	デジタルコピー機の人工イベントログ	XES	2012
30 アクティビティを持つ構造化制御フローからランダム生成された人工イベントログ		XES	2012
ローン申請プロセスログ (調整可能プロセスモデル発見の各種アプローチのテスト用)		XES	2013
BPM 2013	大規模ペトリネットモデルの適合性検査	mxml	2013
大規模銀行トランザクションプロセス		XES, pnml	2014
IS 2014	単一入口/単一出口 (SESE) 分割化適合性検査	mxml, tpn	2014
プロセスドリフトを含む 72 個のローン査定プロセスに基づくイベントログ		mxml	2015

表 2: 3TU.Datacentrum に登録されている, 合成イベントログ (Synthetic Event Logs)

製造プロセス処理ログデータ	CSV	2014
---------------	-----	------

表 3: 3TU.Datacentrum に登録されている, その他のログデータ

ができる。実際のところ、MiMo は例題となるプロセスモデルを含めたプロセスとして配布されている。このことから、MiMo ではマイニングアルゴリズムの実験と改良に向いている²⁵。

2.6 CPN tools

本サーベイの第1回でも紹介した CPN tools^{26 [34]} は、図的編集機能とシミュレーション機能を備えた色付きペトリネットのモデリング・ツールである。したがって、MiMo における ExSpect と同様に、図的エディタで構築した任意のプロセスモデルからログを生成することができる。但し、実運用データではないので、生成したログ中に、現実によく使われるであろうトレースが頻出するわけではない。そのためトレースの出現頻度に関わるようなアルゴリズムの評価に利用することは適切ではない。

ここでは、文献^[35, 36]にしたがって、手順を簡

単に説明する。ProM CPN Library²⁷より、cpnlibrary.zip ファイル²⁸をダウンロードし展開する。CPNToolsConverter フォルダ中の MXMLlogs サブフォルダの中にある loggingFunctionsMultipleFiles.sml は、CPN で記述されたプロセスからイベントデータを抽出しファイルに保存するための ML 関数群を定義している。CPN で記述した対象プロセスモデルを加工し、トレース取得を開始する前のトランジションでケース (事案) 毎のログファイルを初期化する関数 createCaseFile を呼び出した後で、実行トレースを構成するイベントが発生するトランジションにおいて、addATE(ATE は Audit Trail Entries の略) 関数を呼び出すように手を加えれば、ログファイルに記録を残すことができる。

具体例は、同梱されている cpnToolConverter-MXML.cpn にある。これは、ログデータを残すように加工された罰金処理 (Fine Handling) のプロセスモデル定義である。まず、必要なケース (事案) の個数だけプロセスモデルを駆動するトーク

²⁵但し、ExSpect の開発は 2000 年のリリース以降、開発が停止しており、試用ライセンスでデモ版を試用できるだけである。Windows-10 での実行も可能であるが、稀に動作が不安定となるので実用には向いていない。

²⁶<http://cpntools.org/>

²⁷<http://www.processmining.org/tools/promcpn>

²⁸http://www.processmining.org/_media/tools/cpnlibrary.zip

ンを生成し²⁹, (トークンに与えた時間変数により)生成し終わってから,それを,プロセスモデルへ送り込む.このプロセスモデルはトランジションの記述 (inscription) 中で,上記の ML 関数を呼び出している.したがって,この CPM モデルをシミュレーション実行すると³⁰,大域変数として定義されている FILE 変数と FILE_EXTENSION の値に基づき,ログファイルの名前と場所を決定し,指定した情報をログファイルに書き込む.

図 1 中の赤いプレースよりも上の部分がケース ID 付きのトークン生成器であり,赤いプレースよりも下の青く着色したモデルが,対象となる罰金処理のプロセスモデルである.

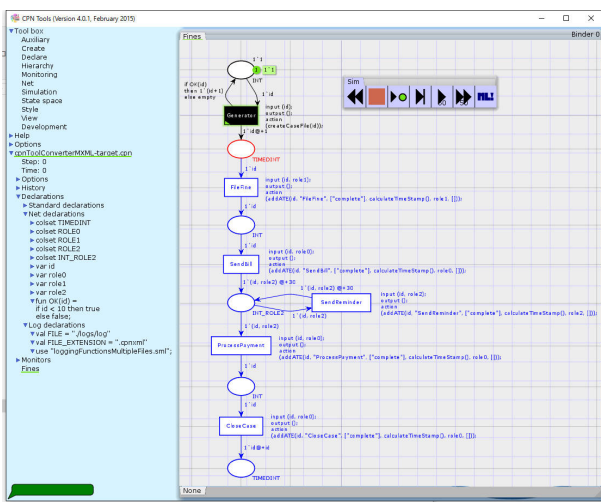


図 1: CPN Tools によるログデータ生成

この方法での生成したログは,後述する ProM へ読み込みこともできるが,ログ生成にあたり,ある程度は色付きペトリネットに対する知識も求められるため,色付きペトリネットに関する日本語の解説が決して多くない現状では,少々,敷居が高いかもしれない.

2.7 その他

このようにモデルのシミュレーション実行によってトレースを生成することは,モデルの性能解析のためにも行われる(ベンチマーク生成)^[37].

²⁹ トークン数は大域宣言している OK 関数の中で指定する
³⁰ CPN Tools の GUI は独特なため慣れないと迷うかもしれない.モデルをシミュレーション実行するには,CPN tools を起動した後,左側のナビゲーションから,ツールボックスのシミュレーションを,ワークスペースにドラッグ&ドロップすると,実行(連続実行やステップ実行)や停止のための一連のボタン群が表示される.

ベンチマーク生成は,ProM のプラグインとしても取り入れられている^[38].

また,確率的文脈自由文法 (stochastic context-free grammar; SCFG) を用いたログ生成ツール. PLG(Processes Logs Generator)³¹^[39], PLG2³²^[40], 宣言的プロセスモデル,すなわちタスク間の制約関係集合である Declare モデル(第 3 節)から正規表現 (Regular Expression) に変換し有限状態機械 (FSM) でシミュレートするログ生成ツール³³^[41], オントロジに基づくデータアクセス (ontology-based data access; OBDA) パラダイムに基づく OnTop システム³⁴ を基礎とした ProM プラグイン (Ontology-Driven Extraction プラグイン)³⁵^[42] もある.これらについては次回(第 3 回 データ編)で触れる予定である.

ログは一般的にモデルにとっての正例 (positive example) といえるが,人工的に生成した負例 (negative example) によってマイニングされるモデルの品質を高める AGNEs(Artificially Generated Negative Events) 手法³⁶^[43] もある.これは,利用者が与えた負例を用いて帰納論理プログラミング (ILP; Inductive Logic Programming)³⁷ を用いた学習によりモデルを生成する手法^[44, 45, 46] の一つとして,次々回(第 4 回 アルゴリズム編)で言及する予定である. AGNEsMiner³⁸ のアルゴリズムを基礎とした ProM へのログ生成プラグイン (Straightforward Petri Net-based Event Log Generation) もある³⁹^[47].

3 ProM

3.1 ProM とは

ProM(Process Mining Framework) は,プロセスマイニングのための総合的な環境であり,オープンソースソフトウェアとしてリリースされてい

³¹ <http://padova.processmining.it/sw>

³² <http://plg.processmining.it/>

³³ <https://github.com/processmining/synthetic-log-generator/>

³⁴ <http://ontop.inf.unibz.it/>

³⁵ <http://ontology-driven-extraction.inf.unibz.it/>

³⁶ <http://perswww.kuleuven.be/~u0041863/AGNEs.php>

³⁷ ILP という略称は,文脈によって,帰納論理プログラミング (Inductive Logic Programming) ではなく,整数線形計画法 (Integer Linear Programming) を指すこともあるので注意してほしい.

³⁸ <http://www.processmining.be/agnesminer/>

³⁹ <http://processmining.be/loggenerator/>

る。ProMは、プロセスマイニングのユーザと、アルゴリズムの開発者の、双方のためのプラットフォームである。ユーザに対しては多種多様なアルゴリズムがプラグインとして提供される。また、開発者には、独自のアルゴリズムをProMのプラグインとして利用する機会を提供する。この双方にアプローチすることで、ProMはプロセスマイニングプラットフォームのデファクトスタンダードを目指している。

とは言うものの、実質的にはアルゴリズムの概要を理解していないと使いこなすことが難しく、まだ一般ユーザ向けのプロダクトと呼べるものではない。そこで、エンドユーザ向けのProM Liteというリリースが公開されている。ただし、ProM Liteのプラグインはアップデートが可能となっているため、刊行物から参照することができないという制約はある。本稿では、LiteではないProMのリリースについて解説する。

3.2 インストール

ProMはJavaで開発されているため、多くのプラットフォーム上で動作する。自身の環境に適合したバイナリを取得していただきたい(図2)。

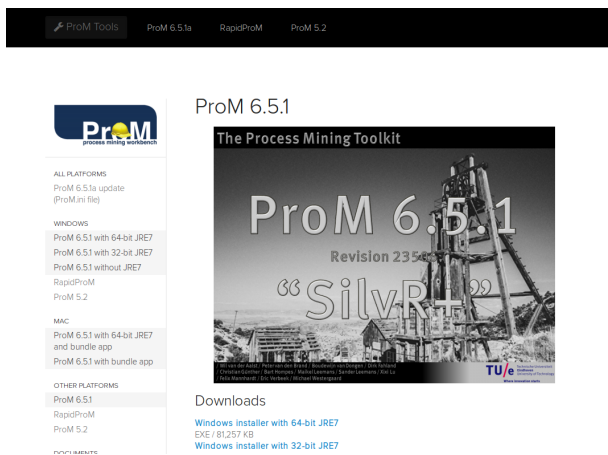


図2: ProMのダウンロードサイト

本稿の執筆時点ではProM 6.5.1が最新のバージョンであり、Windows用(JRE7 64bitあり, JRE7 32bitあり, JRE7なし), Mac用, その他2種類の計6種類が配布されている。ダウンロードされるバイナリにはご注意ください。

3.3 ProMの起動

Windowsでの起動は、インストールされたアイコンをクリックするだけと簡単になっている。Macでもbundle化されているため、アプリケーションのアイコンをクリックすればよい。Linux等の環境では、シェルスクリプトを実行して起動する必要があるが、いずれにせよ非常に容易である(図3)。

なお、初回の起動時にはプラグインのダウンロードが大量に行われるので注意されたい。よく使うアルゴリズムは表4の通りであるので、選択してインストールすることをお勧めする。



図3: ProMの起動画面

3.4 Exerciseデータの利用

ProMの入力形式はXESファイルという標準であり、まずはこのXESファイルを用意しないと簡単な動作確認もできない。幸い、プロセスマイニングの論文でも利用されているような簡単なログの例が、XESファイルとして入手できる(<http://www.promtools.org/prom6/downloads/example-logs.zip>)。ここではexercise1.xesを例に、 α -アルゴリズムを適用するまでの手順を解説する。

まず初めに、XESファイルを開いて入力ソースとして指定する(図4)。

ここで指定したファイルには、A, B, C, D, Eのタスクが含まれ、それらが図5の順で登場する。

次に、入力に使用するプラグインを選択する。ここではNaiveを選択している(図6)。

一連の処理を行うと、exercise1.xesのデータが入力され、インポート済みの状態になる(図7)。

表 4: ProM でプラグインとして提供されているプロセス発見アルゴリズムの例

アルゴリズム	プラグイン
α -アルゴリズム	Mine for a Petri-net using Alpha-algorithm
Heuristics Miner	BPMN Analysis (using Heuristics Miner)
Fuzzy Miner	Mine for a Fuzzy Model
Genetic/Evolutionary	Mine a Process Tree with ETMd

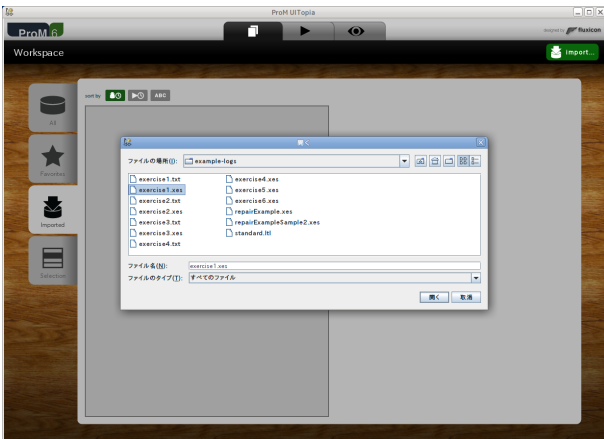


図 4: ProM での XES ファイルのオープン

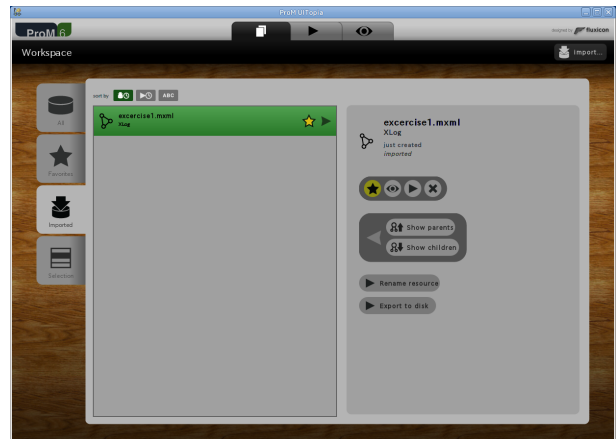


図 7: ProM でのデータ入力完了状態

ABCD
ACBD
AED

図 5: ProM に入力するファイルの内容

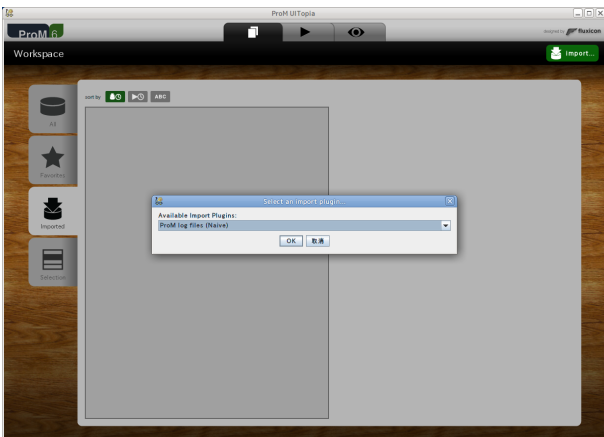


図 6: ProM での入力プラグインの選択

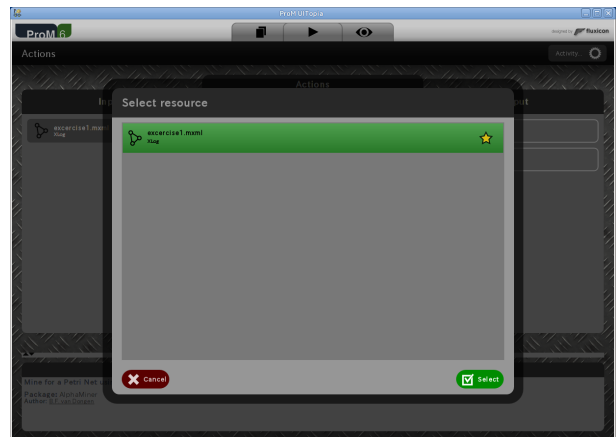


図 8: ProM での入力ソース選択状態

ここで、今回使用する α アルゴリズムを選択する。アルゴリズム名は Mine for a Petri Net using Alpha-algorithm である。見つからなければ検索を利用することもできる。その後、入力ソースの選択を行う (図 8)。

すると、アルゴリズムのアイテムが赤から緑に変化し、マイニングが可能な状態となる (図 9)。

ここで Start を押すとマイニングが実行され、結果が表示される (図 10)。今回表示されたのは、exercisel.xes から求められた Prtri-net である。

ここで +complete と表示される理由は、プロセスマイニングの文化によるものである。プロセスマイニングでは、イベントの終了と開始を明確に分ける。しかし、今回使用したデータでは A, B, C, D, E といったように開始と終了が分かれていない。そこで、便宜的に、.xes ファイルで終了の属性を各イベントに持たせている。

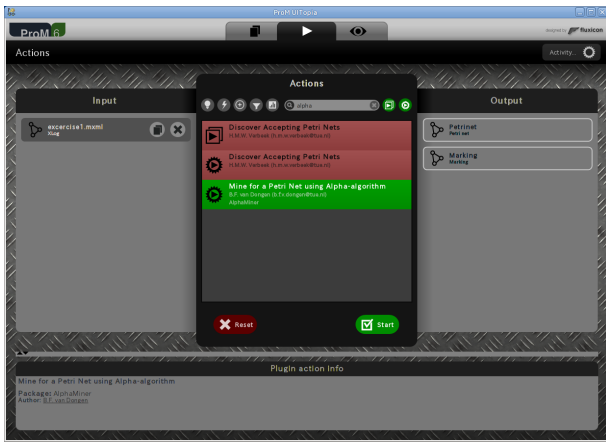


図 9: ProM でのマイニング実行前

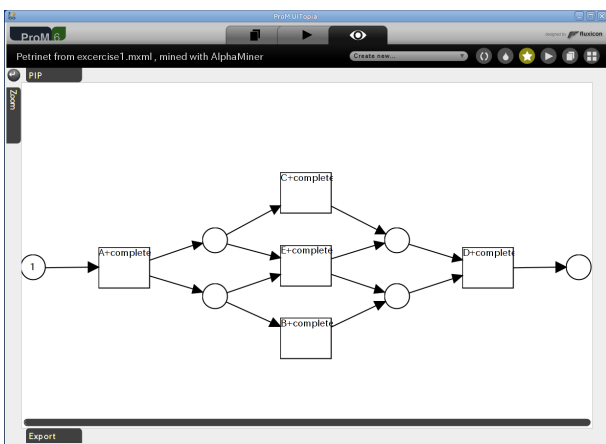


図 10: ProM でのマイニング実行後

4 初期のツール

この節では、文献には登場するが、すでに開発が停止しているツールのいくつかに関して（特に現時点でも試用することができるものを中心に）、概要と相互の関係について述べる。

具体的には、Eindhoven University of Technology で開発された、MiMo, EMiT, および Little Thumb と、それらの関係ならびに、やはり初期のツールである Process Miner や InWoLvE との比較に関して述べる。

4.1 MiMo

MiMo(Mining Module)⁴⁰ [3, 4] は、高水準ペトリネット（階層的 WF-net）に基づく離散プロセ

⁴⁰<http://www.processmining.org/discontinued/mimo>, 本稿執筆時点では開発は停止しているが試用可能である

スモデリングツールである ExSpec(EXecutable SPECification Tool) [48] 上で構築されたツールボックスである。

ExSpec におけるプロセスモデルである階層的 WF-net は、メッセージパッシングで通信しあうサブタスクの集まりであり、さらにサブタスクがプロセスモデルを持つことで階層性を有する。ExSpec は、階層的 WF-net のための図的エディタとシミュレータ（連続実行や、ステップ実行、デバッガのようなブレークポイント設定などができる）を備え、プロセスモデルの構造的正当性 (structural correctness) 解析のために使うことができる。

MiMo は、ワークフローログ生成器 (log generator) とワークフローログ解析器 (log analyzer) をもち、プロセスマイニングアルゴリズムの実験を実施することが可能である。

ログ生成器は、ExSpec の図的エディタでモデル化されたプロセスモデルからワークフローログを生成することができる。ワークフローログは、ケース (事案) 毎の実行トレースの集合である。実行トレースは、本来はイベントの時系列であり、タスクの実行には時間経過を伴うので、一般的には開始イベントと終了イベントの二つのイベントが関与するが、ここでは簡単のために、タスクの終了イベントにのみ注目する、もしくは、タスクの実行に実行時間を伴わないと仮定することで、実行トレースを実行されたタスク ID の時系列として扱う⁴¹。

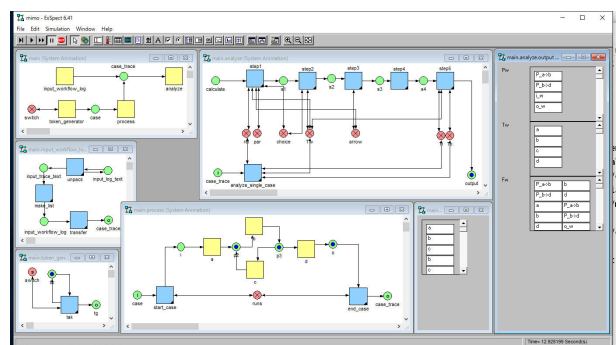


図 11: MiMo での実行例 (中央下がログを生成させるプロセス, 中央上が α -アルゴリズム)

⁴¹但し、二つのタスクが同時に実行されたとしても、実行トレースでは直列化して、どちらかが先に記録されて、タスク間には全順序が与えられる。したがって、イベント情報にタイムスタンプのような属性情報が含まれていなければ、1 トレースだけでは、並行に実行されたのか、順に実行されたのかという区別はつかない

実際の情報システムを構成するサービスでは、多くの場合、ケースごとに逐次実行(実行中の一つのケースが終了するまで次のケースが開始されない)するものではなく、マルチスレッド処理によって複数のケースを同時並行的に実行する。したがって、ログには、ケースごとの実行トレースだけでなく、複数のケース毎の実行トレースが混合して記録されるが、ケースIDによって個々のケースに分離することができる。

ワークフローログ解析器は、プロセスマイニングを行う。MiMoでは、プロセスマイニングのアルゴリズムとして α -アルゴリズムを実装しており、ログデータからモデル(ExSpectで扱える階層的WF-net)を構築することができる。 α -アルゴリズムの詳細は第4回で解説するが、簡単に言えば、イベント相互間の順序関係を収集し推論する手法である。

ログ生成器とログ解析器を備えることにより、与えられたログデータからログ解析器でモデルを構築するだけでなく、図形エディタで構築したモデルからログ生成器でログデータを生成し、さらにそのログデータからログ解析器でモデルを再構築することもできる。

α -アルゴリズムを実装しているMiMoは、 α -アルゴリズムを拡張したアルゴリズムを実装しているツールであるEMiTとLittle Thumbの前身にあたる(後述)。

ExSpect⁴²は、Eindhoven University of Technologyで開発が始められ、Bakkenist consultancy (Deloitte & Touche Bakkenist)に継承された有償ツールである。2000年にリリースされた最終バージョン(6.41)以降は開発がストップしており新たなライセンス発行もされていないが、本稿執筆時点では、ダウンロードも可能であり、試用ライセンスを用いて動作を試すことは現在でも可能である⁴³。

⁴² <http://www.exspect.com/>, <https://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db/exspect.html>

⁴³ α -アルゴリズム自体は、前述のようにProMで実行することができることから、MiMoを実用目的で使うことは現在では推奨されていない

4.2 EMiT

EMiT (Enhanced Mining Tool)⁴⁴ [5, 6, 7] は、 α -アルゴリズムの欠点であった、短いループ(長さ1もしくは2のループ)検出を効率よく取り扱うことができるようにした α^+ -アルゴリズム [49, 50] を実装しており、SWF-ネット(構造化WF-ネット)を発見することができる。 α^+ -アルゴリズムの詳細は第4回で解説するが、簡単に言えば、 α -アルゴリズムの前後に、前処理と後処理を行い、短いループ(長さ1もしくは2のループ)検出に対応する手法である。

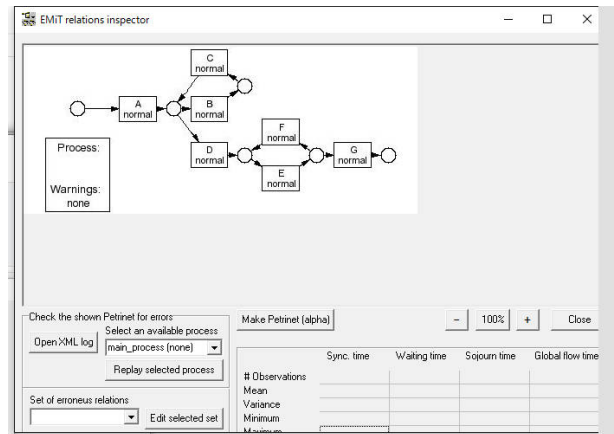


図 12: EMiT の実行例 (原子的タスク)

また、EMiTではタイムスタンプを取り入れており、経過時間を伴わない原子的(atomic)タスクを前提としていない。すなわち、タスク毎に、開始(scheduled)時刻と完了(complete)時刻を取り扱うことができ、あるタスクが完了する前に、別のタスクが開始されたなら、両タスクの間に並行性を見出すことができる。

4.3 LittleThumb

Little Thumb⁴⁵ [8] は、ノイズが混入したログや不完全なログに対応できるように α -アルゴリズムを拡張したヒューリスティック(heuristic)アプローチ [51] を実装したものであり、依存関係/頻度表(dependency/frequency table)および、依存

⁴⁴ <http://www.processmining.org/discontinued/emt>, 本稿執筆時点では開発は停止しているが試用可能である

⁴⁵ <http://www.processmining.org/discontinued/littlethumb>, 本稿執筆時点で開発は停止しているが試用可能である

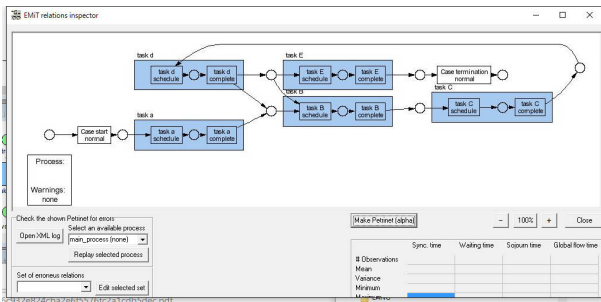


図 13: EMiT の実行例 (タスクの開始時刻と完了時刻の取り扱い)

関係/頻度グラフ (dependency/frequency graph) を用いる。

このヒューリスティック・アプローチは、ProM のプラグインとして、ヒューリスティック・マイナー (Heuristic Miner) に受け継がれているので、ProM をつかってアルゴリズムを試すことができる。ヒューリスティック・アプローチのアルゴリズムに関しては、本サーベイでは、第 4 回のアルゴリズム編で解説する予定である。

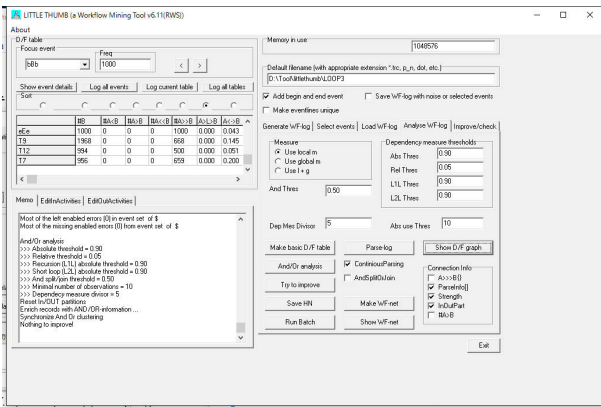


図 14: Little Thumb の実行例

4.4 初期のツールの比較

文献 [9] にしたがって、初期のツールの比較について述べる (表 5)。この評価基準から、初期のツールにおいて何が課題であったかという点に気づくことができる。そうした課題に基づいて、アルゴリズムの拡張がなされ、新たなツールが開発されている。すなわち、ツールの比較というだけでなく、各ツールが実装しているアルゴリズムの比較という意味合いが強いものとなっている。

Priocess Miner のみ「構造」項目がブロックと

なっているが、これは、構造化されたモデルを対象としていることを意味している。EMiT と Little Thumb を比較した場合、EMiT では「時間」項目に○がついているが、これは、原子的 (atomic) タスクだけでなく、時間経過を伴うタスクとして、タスクの開始や完了といったライフサイクル⁴⁶を意識していることも意味している。また、Little Thumb では、「ノイズ」項目に○がついているが、これは、EMiT の実装している α -アルゴリズム (α^+ アルゴリズム) に対して、ヒューリスティックアプローチというアルゴリズム上の拡張に対応している。また、「非自由選択 (non-free choice)」という項目への対応は、 α^{++} アルゴリズムへの拡張につながっている。

さらに、この基準を元に拡張して、タイプの制御フロー発見の多様なアルゴリズムを、大きな分類カテゴリごとに特徴づけることを試みた調査報告 [52, 53] もある。その内容に関しては第 4 回のアルゴリズム編で触れる。

5 その他のオープンソースのツール

ここでは、入出力の形式が前出のツールとは少し異なる 3 つのツールについて紹介する。

5.1 VipTool

VipTool⁴⁷は、ドイツの University of Karlsruhe での VIP (Verification of Information Systems by evaluation of Partially ordered runs) プロジェクトにおいて開発されたツールであり、ペトリネットのモデル化、シミュレーション、検証の機能をもつ。

VipTool では、通常のプロセスマイニングで使われる実行トレース (実行されたタスクないしアクティビティ名の時系列) の代わりに、半順序構造をもつ実行系列 (partially ordered run) を用い、これをシナリオと呼ぶことがある。半順序構造を用いるのは、全順序化した実行トレースでは並行性の情報が失われるが、そのことが離散システムにとってはふさわしくないという判断に基づく。半順序構造をもつ実行系列 LPO (Labeled Partial Order) は、グラフ構造としては有向非巡

⁴⁶ 次回のデータ編で詳述する

⁴⁷ https://www.fernuni-hagen.de/sttp/forschung/vip_tool.shtml, 現在のホームページは、Jörg Desel の所属する FernUniversität Hagen, Distance University of Hagen にある。

表 5: 初期のツールの比較 (文献 [9] より翻訳転載)

	EMiT	Little Thumb	InWoLvE	Process Miner
構造	グラフ	グラフ	グラフ	ブロック
時間	○	×	×	×
基本的な並行性	○	○	○	○
非自由選択	×	×	×	×
基本的なループ	○	○	○	○
任意のループ	○	○	×	×
隠れタスク	×	×	×	×
重複タスク	×	×	○	×
ノイズ	×	○	○	×

回グラフ (DAG; Directed Acyclic Graph) で表現でき、VipTool の図的エディタで構築して、lpo という拡張子を持つファイルに保存することができる。VipTool は、さらに、こうして構築した複数の LPO からペトリネットを合成する機能をもつ。

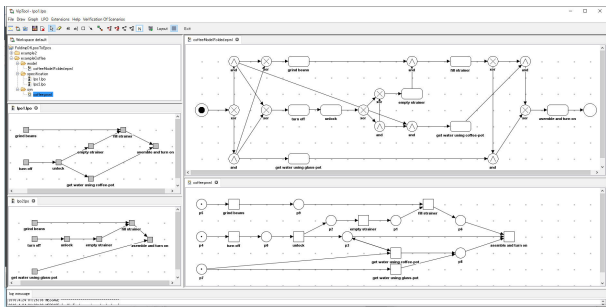


図 15: VipTool の使用例 (1) 左下の二つが実行系列 LPO, 右上はマイニング対象のプロセスモデル, 右下はマイニング結果のモデル

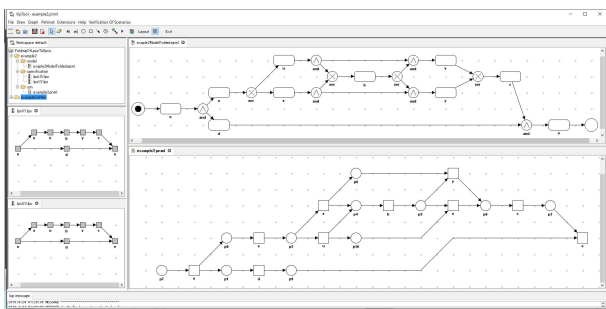


図 16: VipTool の使用例 (2) 左下の二つが実行系列 LPO, 右上はマイニング対象のプロセスモデル, 右下はマイニング結果のモデル

5.2 Declare

5.2.1 宣言的プロセスモデル

Declare システム⁴⁸ [17, 18, 19] は、宣言的プロセスマイニング (Declarative Process Mining) [16] のツールである。Declare システムで用いる Declare モデル⁴⁹では、プロセスの振る舞いは、そのプロセスが実行されている間に満たすべきルールの集合として記述される。これにより、マイニングされたプロセスの大規模化や複雑化を避け、関心のある性質に限定して注目したマイニングプロセスの支援を実現することを目指している。

必ずしも (プロセスマイニングの出力である) 制御フローを直接的に表現するのではなく、記述者が必要と考えたタスク間の制約関係のみを記述することから、その制約条件を満たす制御フローが存在しなかったり (もちろん論理的関係を宣言しているので整合性をチェックすることができる)、逆に制約を満たす制御フローが一意に特定できないこともありうる。このことから、前回のモデル記述言語の解説においては、あえてプロセスモデルのためのモデリング言語として紹介せず、今回、プロセスマイニングのための入出力の一部として取り扱っている⁵⁰。

宣言的プロセスモデルは、動的変更に強い柔軟なワークフロー管理システムを実現するものであるが、プロセスマイニングの文脈では、マイニングの出力としてのプロセスモデルとしても、入力

⁴⁸ <http://www.win.tue.nl/declare/>

⁴⁹ 以前は、ConDec と呼ばれていたモデリング言語とその言語で記述されたモデルのいずれも、支援ツールである Declare システム (従来は大文字表記の DECLARE システム) の名称に準じて、それぞれ、Declare 言語、Declare モデルと呼んでいる模様である。本稿では、それ以前の第 2.2.0 版の Declare システムで動作確認している

⁵⁰ とはいっても、宣言的プロセスモデルも、もちろん広い意味で、プロセスをモデル化しているものに他ならない

データであるログに対して補足的な知識を付け加えるための入力としても使うことができる。

Declare では、タスクの間の時間的な制約関係を、線形時制論理 LTL(Linear Temporal Logic) で与えるが、図的に表現することができる。LTL で使用できる演算子は、通常の(命題論理の)論理演算子である、含意(\Rightarrow)、論理積(\wedge)、論理和(\vee)、否定(\neg)に加えて、表 6 の時制様相演算子が加わる。

表 6: 線形時制論理 LTL で使用する時制様相論理演算子

つねに	$\Box p$	将来のすべての状態で p が成り立つ
いつかは	$\Diamond p$	将来のいつかの状態で p が成り立つ
次の状態で	$\bigcirc p$	次の状態で p が成り立つ
までは	$p \sqcup q$	q が成立するまで p が成立する

一つ一つの制約関係は、タスク(図 17 中で四角形であらわされている)の間を結ぶ線分で表現され、図形的エディタではテンプレートによって定義することができる。図 17 中の左側のナビゲーションメニューからテンプレートを選択し、テンプレートのプレースホルダとなっているタスクを具体的なタスクで置き換えていくことで、モデルを構築できる。

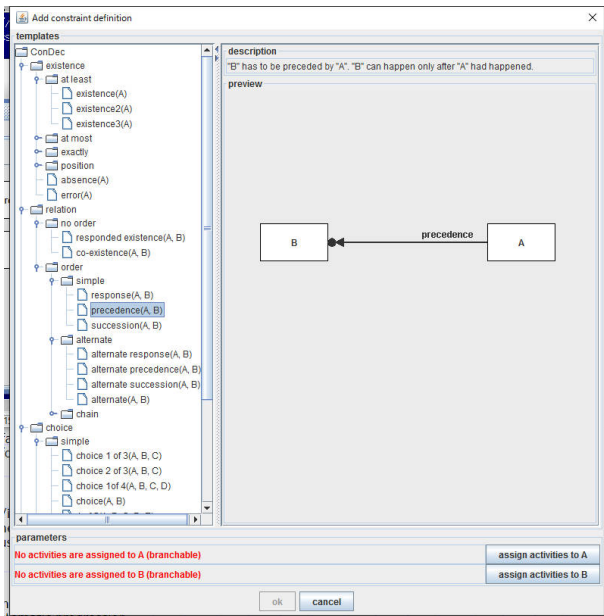


図 17: Declare

タスク並びにタスク間与えられる制約条件は、大きく以下のように分類できる。

- (1) タスク間の関係テンプレート(表 7)

- (2) タスク間の否定的関係テンプレート(表 8)
- (3) タスクの存在性テンプレート(表 9)

図 18 に示すように、Declare モデルでは、タスク間の制約関係を図示することで、個別に与えられた制約関係を、図的なモデルとして総合的に俯瞰することができる。

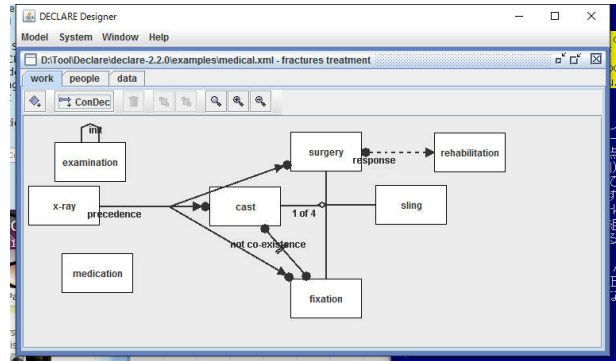


図 18: Declare におけるタスク間の制約関係

この宣言的モデル(Declare 言語)は、Declare システムだけでなく、広く使われてきており、前出の色付きペトリネットのモデリングツール CPN Tools にも、Declare モデルが統合されつつある⁵¹

5.2.2 宣言的プロセスマイニング(Declare Maps Miner)

宣言的プロセスマイニングのツールとしては、ProM6.1 以降では、ProM のプラグイン Declare Miner⁵² が Declare パッケージに含まれている [20]。

ここでは、第 3 節で導入した ProM 6.5.1 の Declare Maps Miner によって、ログデータから Declare モデルをマイニングする様子を示す。前出の Exercise データ(exercise-logs.zip)の中から、exercise4.xes を用いる。exercise4.xes の内容に対応する exercise4.txt の内容を図 19 に示す。1 個以上の空白文字で区切られたテキストファイルになっているが第 1 列はトレースの出現頻度、第 2 列はケース ID を意味しており、第 3 列から行末までが実行トレースである。このトレースでは、原子的なタスクを扱っているので、XES 形式に変換する際に、タスクの完了時刻(complete)を採用している。

⁵¹2012 年の CPN Tools 3.5.1 版から、本稿執筆時の最新版である 4.0.1 版(2015 年 2 月)に至るまで統合作業が進められているが、未だ動作は、必ずしも安定していない模様である。

⁵²<http://www.win.tue.nl/declare/declare-miner/>

表 7: Declare モデル (1) タスク間の関係テンプレート

名称	制約	Declare 表現
responded existence(A, B)	$\diamond A \Rightarrow \diamond B$ A が (少なくとも 1 回) 実行されるならば, B も (少なくとも 1 回) 実行されなければならない	
co-existence(A, B)	$\diamond A \Leftrightarrow \diamond B$ A が (少なくとも 1 回) 実行されるならば, B も (少なくとも 1 回) 実行されなければならない, その逆も成り立たねばならない	
response(A, B)	$\square(A \Rightarrow \diamond B)$ A が実行されるときはいつでも, その後で B もいつかは実行されなければならない	
precedence(A, B)	$(\neg B \sqcup A) \vee \square(\neg B)$ B が実行されるなら, その前に A も実行される.	
succession(A, B)	$response(A, B) \wedge precedence(A, B)$ response かつ precedence: 1. 各 A の実行後には少なくとも 1 回 B が実行される. 2. B が実行されるなら, 先行して A も実行されなければ ならない. B は A の後にのみ実行できる	
alternate response(A, B)	$\square(A \Rightarrow \bigcirc(\neg A \sqcup B))$ 各 A の実行後には, B が実行されるが, その間に別の A が実行されることはない	
alternate precedence(A, B)	$prec.(A, B) \wedge \square(B \Rightarrow \bigcirc(prec.(A, B)))$ B が実行されるときは, その前に A が実行されるが, その間に別の B が実行されることはない	
alternate succession(A, B)	$alt.response(A, B) \wedge alt.precedence(A, B)$ Alternate response かつ alternate precedence: 1. 各 A の実行後には, 少なくとも一回 B が実行されるが, その間に別の A が実行されることはない. 2. B が実行されるときは, その前に A が実行されるが, その間に別の B が実行されることはない	
chain response(A, B)	$\square(A \Rightarrow \bigcirc B)$ A の実行の次の状態で B が実行されなくてはならない	
chain precedence(A, B)	$\square(\bigcirc B \Rightarrow A)$ B は A の実行直後にのみ実行できる	
chain succession(A, B)	$\square(A \Leftrightarrow \bigcirc B)$ A と B は互いの次の状態にのみ実行できる	

表 8: Declare モデル (2) タスク間の否定的関係テンプレート

名称	制約	Declare 表現
not co-existence(A, B)	$\neg(\diamond A \wedge \diamond B)$ A か B のうち一方のみが実行でき, 両方は実行できない	
not succession(A, B)	$\square(A \Rightarrow \neg(\diamond B))$ A が実行されたら, その後, B は実行できない	
not chain succession(A, B)	$\square(A \Rightarrow \bigcirc(\neg B))$ A が実行されたら, その直後に, B は実行できない	

表 9: Declare モデル (3) タスクの存在性テンプレート

名称	ラベル	制約	Declare 表現
existence(A)	1..*	$\diamond A$ A は少なくとも 1 回実行されなければならない	
existence2(A)	2..*	$\diamond(A \wedge \bigcirc(\diamond A))$ A は少なくとも 2 回実行されなければならない	
existence3(A)	3..*	$\diamond(A \wedge \bigcirc(\diamond(A \wedge \bigcirc(\diamond A))))$ A は少なくとも 3 回実行されなければならない	
absence(A)	0	$\neg existence(A)$ A は決して実行されない	
absence2(A)	0..1	$\neg existence2(A)$ A は多くて 1 回実行されうる	
absence3(A)	0..2	$\neg existence3(A)$ A は多くて 2 回実行されうる	
exactly1(A)	1	$existence \wedge absence2(A)$ A は正確に 1 回実行されなければならない	
exactly2(A)	2	$existence2 \wedge absence3(A)$ A は正確に 2 回実行されなければならない	
init(A)	init	A 各インスタンスは A の実行で始まらなければならない	

1x Case1 a b c d f
 1x Case2 a c b d f
 1x Case3 a b d c f
 1x Case4 a c d b f
 1x Case5 a d e f
 1x Case6 a e d f

図 19: 例題データ exercise4.txt の内容

この例題データから, Declare Maps Miner でマイニングした Declare モデルを図 20 に示す. a-complete が他のすべてのタスク (イベント) が開始イベントに, f-complete が他のすべてのタスク (イベント) が最終イベントに相当することなどが制約として表現されている.

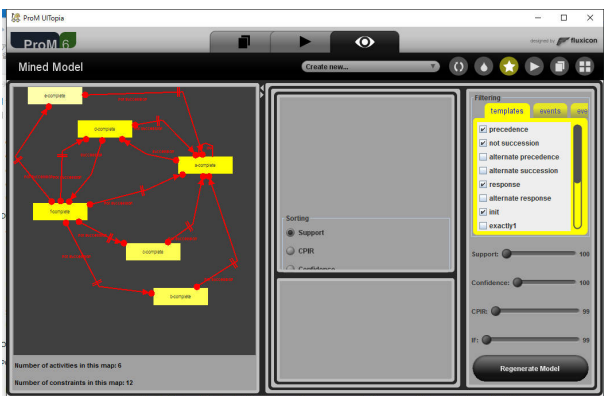


図 20: Declare Maps Miner でマイニングされた Declare モデル

また, すでに Declare モデルが与えられていて,

それに対して, ログをもちいて修正するというアプローチが Repair a Declare Model プラグインで採用されている. これは, ログデータ以前から, プロセスモデルに対しある程度の事前知識があるときに, その知識を Declare モデルで補足補足的に与えた上で, ログデータからマイニングするという考え方で使うこともできる.

5.2.3 宣言的プロセスマイニング (MINERful マイナー)

MINERful アルゴリズム [54] の ProM プラグインであり高速に動作する [55]. 次回紹介する予定の Declare モデルから正規表現へ変換し有限状態機械をシミュレートするログ生成ツール [41] のログ生成部は, この MINERful アルゴリズムの実装⁵³ に含まれていて定量評価に使われている模様である.

同じく exercise4 データから MINERful マイナーでマイニングした Declare モデルを図 21 に示す.

5.2.4 宣言的プロセスマイニング (DecMiner)

ProM5.2 では, ログデータから ILP(帰納論理プログラミング) 手法を適用して Declare モデルをマイニングする宣言的プロセスマイニングのプラグインとして, DecMiner を使うことができ

⁵³<https://github.com/cdc08x/MINERful>

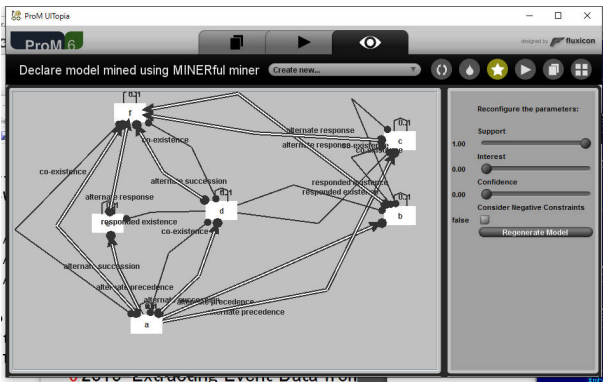


図 21: MINERful マイナーでマイニングされた Declare モデル

る⁵⁴ [56]. このアルゴリズムは、正例と負例を用いることができる点が特徴的である (アルゴリズムに関しては、次々回のアルゴリズム編で解説する予定である).

DecMiner は内部的に論理プログラミング言語 Prolog を試用するため、事前にセットアップしておく必要がある⁵⁵. ログデータは前述の例と同様に exercise4.xes データを MXML データ形式に変換して使用したが、内部的に Prolog の確定節へ変換してから読み込まれるため、その折のパーキングの都合上、一部加工して使用する必要があったことを付記しておく. ログ中の各ケースに対し正例か負例かの分類を行ない (図 22), Declare の制約タイプを選択し (図 23), パラメータを設定した後にマイニングを開始する (図 24).

5.3 Rbminer

Rbminer⁵⁶ [22] は、状態ベースの領域 (Region) 理論に基づくプロセスマイニングツールである. 遷移系 (transition system) として与えられた振る舞いから、プロセスモデルをマイニングすることができる.

同系統のツールとしては、同じグループによる Genet⁵⁷ [21] もあるが、今回は、動作確認の都合上、Rbminer を取り上げる.

Rbminer は、遷移系とペトリネットの表現に

⁵⁴<http://endif.unife.it/it/ricerca-1/aree-di-ricerca/informazione/ingegneria-informatica/process-mining>

⁵⁵動作確認には YAProlog-6.2.2 (<https://www.dcc.fc.up.pt/~vsc/Yap/downloads.html>) を使用した

⁵⁶<http://www.cs.upc.edu/~jcarmona/rbminer/rbminer.html>

⁵⁷<http://www.cs.upc.edu/~jcarmona/genet.html>

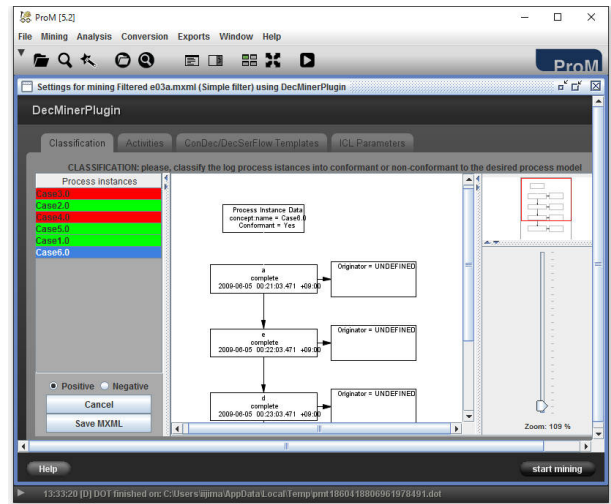


図 22: DecMiner でのログ分類 (正例と負例に分類)

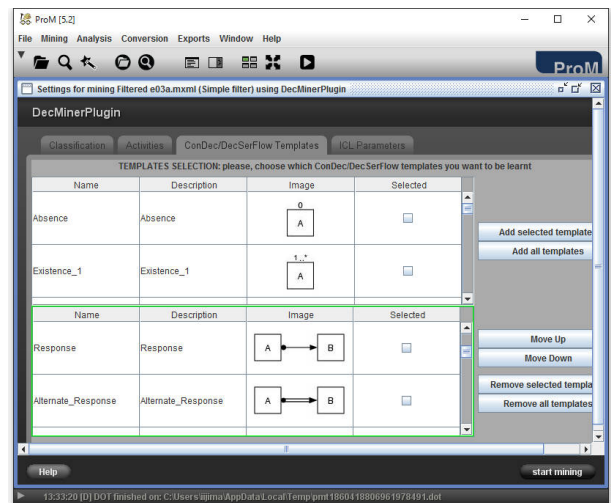


図 23: Declare モデルの制約テンプレートの種類の選択

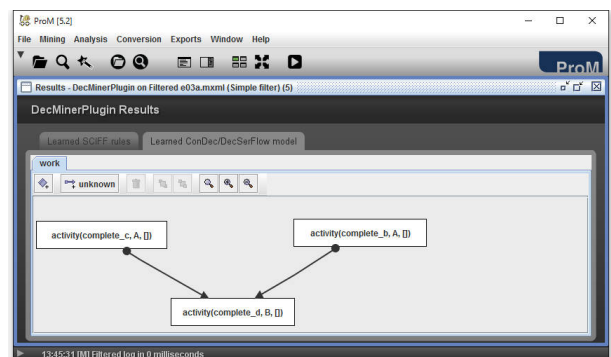


図 24: マイニングされた Declare モデル

SIS 形式^[57]を採用している. そこで可視化には同じく SIS 形式を用いる「ペトリネットと非同期

回路のための合成ツール」Petrify⁵⁸ [58] に含まれる、自動レイアウト機能を備えたdraw_astg コマンド⁵⁹を利用する(具体的には後述)。あるいは、Petrifyをインストールしなくても、Rbminerに同梱されているpntopnml コマンドでPNML(Petri Net Markup Language)形式へ変換しProMにインポートしてもよい(「PNML Reset/Inhiitor net files」プラグイン)。これによって自動レイアウトして表示することができる⁶⁰。

RbminerのLinuxバイナリは入手可能であり⁶¹、コマンドラインで使用することができる⁶²。また、状態グラフやペトリネットの可視化のためにdraw_astg コマンドを使うには、Petrifyのインストールが必要であり、Linuxバイナリが入手可能である⁶³。Rbminerと同じくコマンドラインで利用可能である⁶⁴。

参考までに、利用手順の例を以下に示す。

(1) まず、ログをテキストファイルで用意する。

- ログのデータ形式は、1ケース毎に1行のトレースを記述するもので、トレースは、タスクのラベルを1個以上の空白文字を区切り文字として列挙したものである。
- ここでは、examplesディレクトリに含まれているログデータa12f0n00_1.trを使用する。

(2) 次に、log2ts コマンドでログをSIS形式の遷移系に変換する。

- ログをa12f0n00_1.trというファイル名のファイルとしたとき、
log2ts a12f0n00_1.tr >
a12f0n00_1.ts
とすればよい。

⁵⁸<http://www.cs.upc.edu/~jordicf/petrify/>

⁵⁹astgは、asynchronous signal transition graphを意味する。state transition graphとは異なるものである。また、sg オプションで、state graphを表示することもできる。

⁶⁰ここで用いているstate graphや、pntopnml コマンドで変換されたPNMLファイルにはレイアウト情報が含まれていないため、自動レイアウト機能が不可欠である

⁶¹<http://www.cs.upc.edu/~jcarmona/rbminer/rbminer.html>

⁶²残念ながらソースコードが提供されていないが、Ubuntu 15.10 (64-bit) で問題なく動作することを確認した

⁶³<http://www.cs.upc.edu/~jordicf/petrify/>

⁶⁴実行にあたってはgraphvizのインストールが必要である。また、32bit バイナリなので、64bit 環境で動かすためには、32bit 版ライブラリも追加インストールする必要があるかもしれない。Ubuntu 15.10での実行のためには32bit 圧縮ライブラリlib32z1が必要だった

- SIS形式の状態グラフ(state graph)として、遷移系が出力される。

- 遷移系(状態グラフ)の可視化方法

```
* draw_astg -sg a12f0n00_1.ts
> a12f0n00_1.ps
```

を実行し、PS(Postscript)形式のファイルを生成する。psファイルの表示にはGhostView(gv コマンド)等を使えばよい。

```
* もしくは、ProMにインポートすれば自動レイアウトして表示することができる(「Petrify state graph files」プラグイン)。
```

(3) そして、rbminer コマンドで遷移系からペトリネットを生成する。

```
- rbminer --agg 4 --k 3
a12f0n00_1.ts
> a12f0n00_1.pn
```

を実行するとa12f0n00_1.pn ファイルにSIS形式でペトリネットが生成される。

- --k オプションは、指定した整数で有界なペトリネットを出力することを意味し(デフォルト値は1、すなわち安全ネットである)、--agg オプションは、チェックすべき領域(region)の個数を指定する集約因子(aggregation factor)である(デフォルト値は0、すなわち制限しないことを意味する)。

- マイニング結果のペトリネットの可視化

```
* draw_astg a12f0n00_1.pn
> a12f0n00_1.ps
```

を実行し、PSファイルを生成する。その表示は、遷移系の場合と同様にすればよい。

```
* pntopnml コマンドで
pntopnml a12f0n00_1.pn
> a12f0n00_1.pnml
```

を実行することで、広くペトリネット関連ツールで使われている交換データ形式である、PNMLファイルへ変換することができる。PNMLファイルは、ProMヘイン

ポートすることで自動レイアウトして表示することができる。

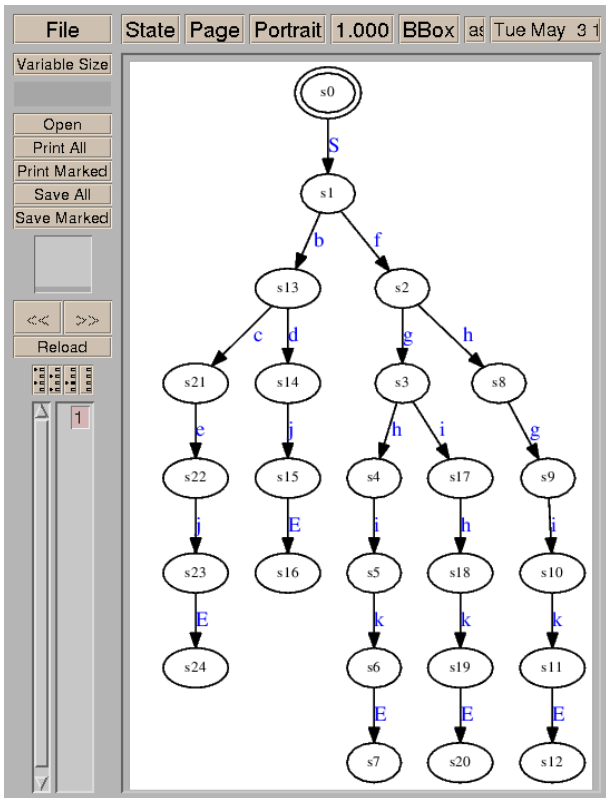


図 25: Rbminer の入力となる遷移系

Rbminer には、上記のコマンド `rbminer`(遷移系からのペトリネットのマイニング)、`log2ts`(ログから遷移系への変換)、`pntopnml`(SIS 形式のペトリネットから PNML 形式への変換) の他に、`create_mxml`(プレインテキストのログから MXML 形式への変換)、`xmltotrace`(MXML 形式のログからプレインテキスト形式へ変換)、`conformancecheck`(ペトリネットがログを満たすことの適合性検査) といったユーティリティコマンドが含まれている。

6 商用のツール

商用ツールとしては Fluxicon 社の Disco⁶⁵ [29] と富士通の Interstage Business Process Manager⁶⁶ を取り上げる。

⁶⁵<http://www.fluxicon.com/disco/>

⁶⁶<http://www.fujitsu.com/jp/products/software/middleware/business-middleware/interstage/products/bpmgr/>

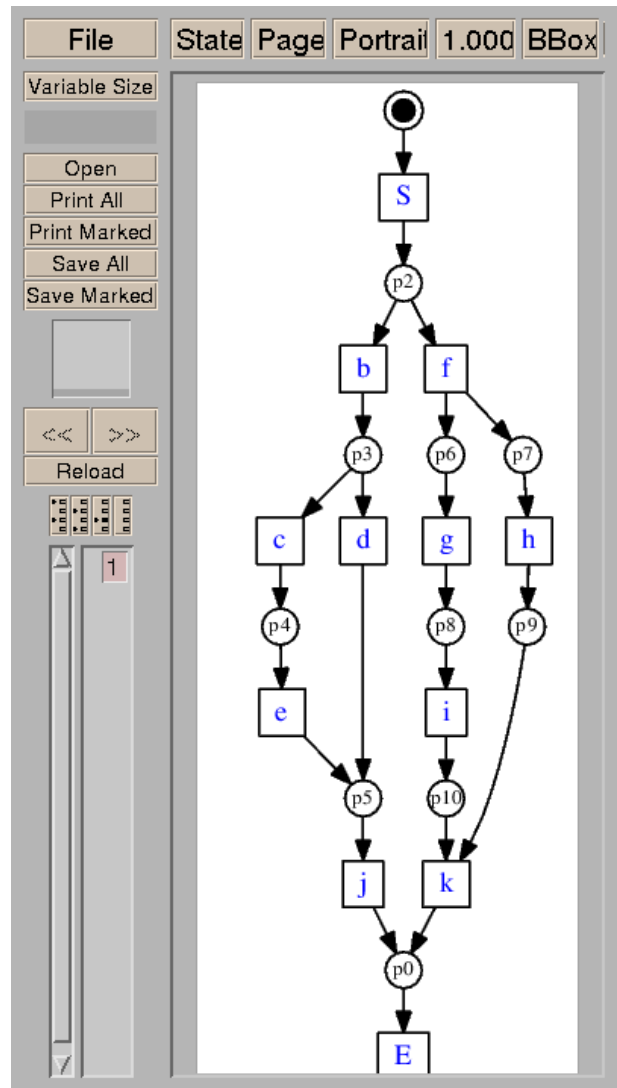


図 26: Rbminer の出力したペトリネット

6.1 Fluxicon Disco

Disco は BPM (Business Process Management) のための商用ツールである。一般的な BPM ではモデルを先に作成する一方、Disco ではプロセスマイニングの研究成果を応用して、ログからボトムアップにモデルを作成することを提唱している。

商用利用するためには商用ライセンスを購入する必要があるが、学生や研究者のために無料のアカデミックライセンスも存在する。

Disco を開発している Fluxicon 社は、共同起業者の 2 名によって営まれているオランダのベンチャー企業である。同社の Rozinat は医療のようなプロセスマイニングの応用に関する研究で、また、Günther は ProM の開発で、それぞれプロセスマイニングの研究開発をリードしてきた。さらに、同社の Advisory Board には、van der Aalst

も名を連ねている。

Disco は、Günther が開発しているだけあって、ProM のファイル形式である MXML 形式と XES 形式の入力にも対応している。

それでは Disco が ProM と違う点は何かという点、第一にマイニングのために独自のアルゴリズムを用いている点、第二に UI/UX の完成度が高い点である。

ここでは、動作例として、Disco に付属する CSV 形式のデータを入力してみる。なお、デモ用ライセンスで利用できるデータは 100 イベントに限られている。本稿ではイベント数に制限のないアカデミックライセンスを使用しているため、読者の手元と結果が異なる可能性がある。

Disco を起動後、ツールチップの指示に従えば、簡単にインポートが完了する。(図 27)

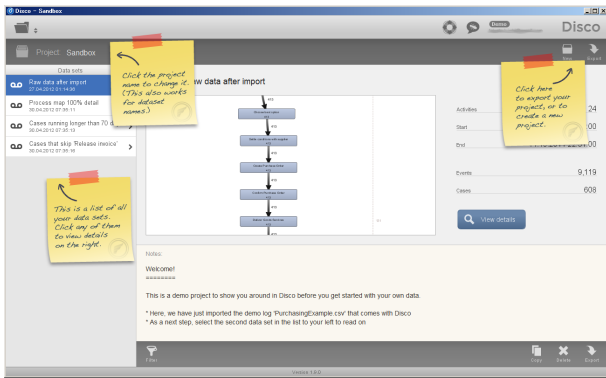


図 27: Disco にデータをインポートしたところ

インポートが完了したあとも、どのような操作が可能であるのか、ツールチップが教えてくれるので、操作に困ることはない。他のツールでは利用方法をユーザが覚えなければならないのに対して、大きなアドバンテージである。このツールチップに沿って操作をしていけば、簡単にマイニング結果が得られる。(図 28)

Disco は、マイニング結果の表示機能がとりわけ優れている。例えば、遷移を表す線の太さでパフォーマンス(経過時間)を表すことができ、ビジネスのボトルネックが一目瞭然となる。このような表示方法は、プロセスマイニングマニフェスト [59, 60, 61] の指針 GP5 でも示されている⁶⁷。

⁶⁷ サーベイ第 1 回でも紹介したプロセスマイニングマニフェストの指針 GP5 「現実の意図的な抽象化としてモデルを扱うべきだ」では、モデルやログ情報の表示に際して、モデルを見る人の目的に応じて、必要な情報を強調し、それ以外の情報を捨象するような視覚効果の導入を示唆している。

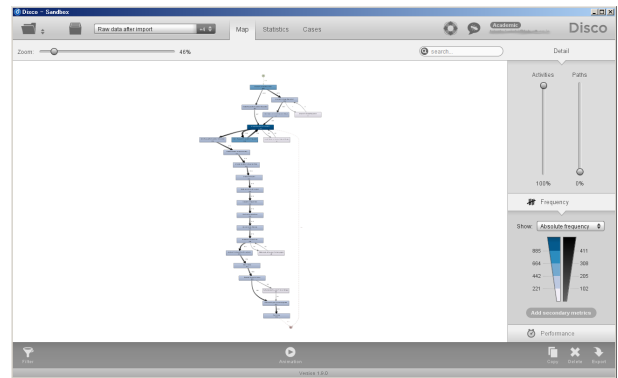


図 28: Disco でマイニング結果を表示したところ

このほか、マイニングされたモデルの上で、イベントをリプレイするアニメーションの機能が搭載されており、イベントの流れを視覚的に把握することが可能である(図 30)。

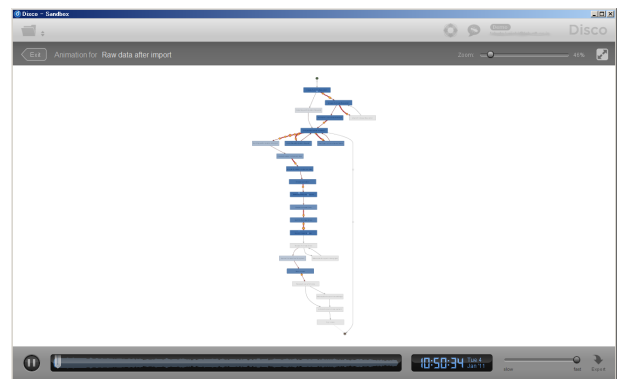


図 29: Disco でイベントをリプレイしたところ

まとめると、Disco は商用製品ならではの完成度を誇る、state-of-the-art なプロセスマイニングツールである。Disco は ProM のファイル形式をインポートすることができるので、ProM を使っている場合は Disco も試してみる価値がある。

6.2 Fujitsu Software Interstage Business Process Manager

Interstage Business Process Manager は、Fujitsu Software Interstage というビジネスアプリケーション基盤の中の 1 つのアプリケーションである。非定型な業務のログを元に可視化することや、BPMN を用いてビジネスプロセスを設計・シミュレートしたり、プロセスのモニタリングを行うことが可能である。BPM という枠組みの中で語られてはいるが、プロセスのモニタリングと

いった、プロセスマイニングのフルセットの機能を有していると言える。

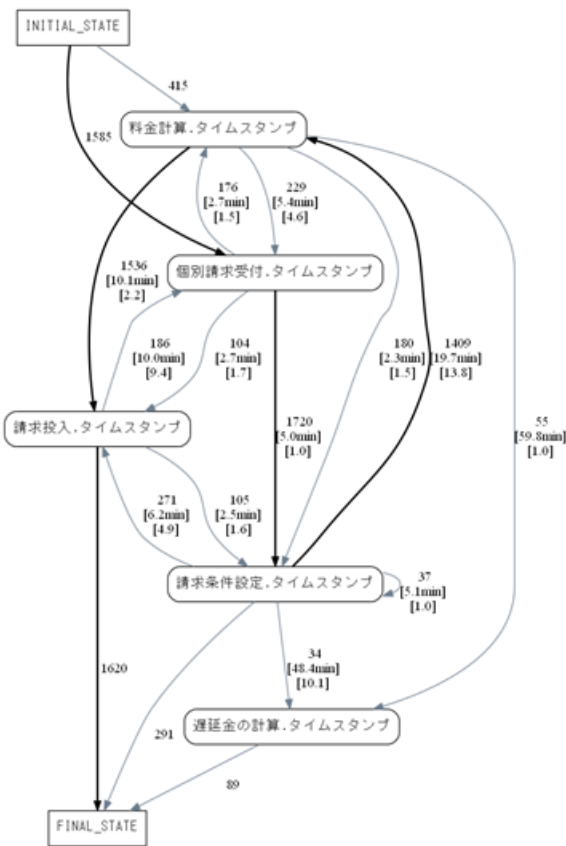


図 30: Fujitsu Interstage Business Process Manager のマイニング結果

7 おわりに

今回は、この先の回で入力データ構造やアルゴリズムの詳細に踏み込むのに先立ち、まずは実際に幾つかのツールに触れて、プロセスマイニング（実際には、そのうちの制御フロー発見）の感触を得ていただくことを目的としている。

加えて、いろいろな文献に現れてくる初期のツールやその相互関係なども理解して、この分野での土地勘を得ていただくために、すでに開発が停止しているツールに関しても（本稿執筆時でも試用できるものを中心に）紹介している。

しかし、いろいろなマイニングアルゴリズム、データ変換ツール、解析ツールは、2004年以降は次第に、ワークベンチとしての ProM のプラグインとして整備されつつあり、第1節で述べたように、最近では、さらにその傾向が強まっている。

したがって、まずは、ProM をインストールし、代表的なマイニングアルゴリズムを試すことをお勧めしたい。第3節は、そうした観点から読者を支援することを目的とした丁寧な記述を心掛けている。

今回も、ツールを試用するという観点から、ツールへの入力データとするワークフローログの入手方法に関して簡単に触れたが、次回（第3回）はデータ編であり、単なるツール試用ではなく実運用に向けて、情報システムの解析目的に合わせて、プロセスマイニングの入力データとしてどのようなデータを与えるべきか、ツールに与えるためのデータの抽出と加工の方法、実際にツールに入力するためのデータ形式と相互変換などに関して解説する予定である。

今回の執筆分担:

本稿の執筆分担は、第1節、第2節、第4節、第5節、および第7節を飯島が担当し、第3節と第6節を田端と斎藤が担当した。全体を通しての調整や補足は飯島が担当した。

文献

- [1] G. Schimm, “Process Miner – A Tool for Mining Process Schemes from Event-Based Data,” *Logics in Artificial Intelligence: 8th European Conference, JELIA 2002*, (eds.) S. Flesca, S. Greco, G. Ianni, and N. Leone, LNAI-2424, pp.525–528, Springer, 2002, http://link.springer.com/chapter/10.1007/3-540-45757-7_47.
- [2] J. Herbst, and D. Karagiannis, “Workflow mining with InWoLvE,” *Computers in Industry*, vol.53, no.3, pp.245–264, 2004, <http://www.sciencedirect.com/science/article/pii/S0166361503001957>.
- [3] W. van der Aalst, A. Weijters, and L. Maruster, “Workflow Mining: Which processes can be rediscovered?,” *BETA Working Paper, WP 74*, 2002, <http://wwis.win.tue.nl/~wvdaalst/publications/p169.pdf>.
- [4] W. van der Aalst, T. Weijters, and L. Maruster, “Workflow Mining: Discovering process models from event logs,” *IEEE Transactions on Knowledge and Data*

- Engineering, vol.16, no.9, pp.1128–1142, 2004, <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1316839>.
- [5] W.M.P. Aalst, and B.F. Dongen, “Discovering Workflow Performance Models from Timed Logs,” Engineering and Deployment of Cooperative Information Systems: First EDCIS 2002, (eds.) Y. Han, S. Tai, and D. Wikarski, LNCS-2480, pp.45–63, Springer, 2002, http://link.springer.com/chapter/10.1007/3-540-45785-2_4, <http://wwwis.win.tue.nl/~wvdaalst/publications/p162.pdf>.
- [6] W. van der Aalst, A. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs,” QUT Technical report FIT-TR-2003-03, Queensland University of Technology, 2003, <http://wwwis.win.tue.nl/~wvdaalst/publications/p195.pdf>.
- [7] W.M.P.v.d.A. Boudewijn F. van Dongen, “EMiT: A Process Mining Tool,” Applications and Theory of Petri Nets 2004, (eds.) J. Cortadella, and W. Reisig, LNCS-3099, pp.454–463, Springer, 2004, http://link.springer.com/chapter/10.1007/978-3-540-27793-4_26, http://www.processmining.org/_media/publications/dongen2004.pdf.
- [8] A.J.M.M. Weijters, and W.M.P. van der Aalst, “Rediscovering workflow models from event-based data using little thumb,” Integrated Computer-Aided Engineering, vol.10, no.2, pp.151–162, 2003, <http://content.iospress.com/articles/integrated-computer-aided-engineering/ica00143>, http://www.processmining.org/_media/publications/weijters2003.pdf.
- [9] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters, “Workflow Mining: A Survey of Issues and Approaches,” Data and Knowledge Engineering, Vol.47, No.2, pp.237–267, 2003, <http://www.sciencedirect.com/science/article/pii/S0169023X03000661>, http://is.ieis.tue.nl/staff/aweijters/WM_overzicht.pdf.
- [10] J. Desel, G. BJuhas, R. BLorenz, and C. BNeumair, “Modelling and validation with viptool,” Business Process Management, BPM 2003, (ed.) A. ter Hofstede, LNCS-2678, pp.380–389, Springer, 2003, http://link.springer.com/chapter/10.1007/3-540-44895-0_26.
- [11] G. Juhas, R. BLorenz, and J. BDesel, “Can i execute my scenario in your net?,” Applications and Theory of Petri Nets 2005, (eds.) G. Ciardo, and P. Darondeau, LNCS-3536, pp.289–308, Springer, 2005, http://link.springer.com/chapter/10.1007/11494744_17.
- [12] R. Bergenthum, J. BDesel, G. BJuhas, and R. BLorenz, “Can i execute my scenario in your net? viptool tells you!,” Petri Nets and Other Models of Concurrency - ICATPN 2006, (eds.) S. Donatelli, and P.S. Thiagarajan, LNCS-4024, pp.380–389, Springer, 2006, http://link.springer.com/chapter/10.1007/11767589_21.
- [13] R. Lorenz, R. Bergenthum, J. Desel, and S. Mauser, “Synthesis of petri nets from finite partial languages,” Seventh International Conference on Application of Concurrency to System Design, 2007. ACSD 2007, pp.157–166, IEEE, 2007, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4276275>.
- [14] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser, “Synthesis of petri nets from finite partial languages,” Applications and Theory of Petri Nets, PETRI NETS 2008, (eds.) K.M. van Hee, and R. Valk, LNCS-5062, pp.388–398, Springer, 2008, http://link.springer.com/chapter/10.1007/978-3-540-68746-7_25.
- [15] R. Lorenz, J. Desel, and G. Juhas, “Models from scenarios,” Transactions on Petri Nets and Other Models of Concurrency VII, (eds.) K. Jensen, W.M.P. van der Aalst, G. Balbo, M. Koutny, and K. Wolf, LNCS-7480, pp.314–371, Springer, 2013, http://link.springer.com/chapter/10.1007/978-3-642-38143-0_9.

- [16] M. Pesic, and W.M.P. Aalst, “A Declarative Approach for Flexible Business Processes Management,” Business Process Management Workshops: BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, (eds.) J. Eder, and S. Dustdar, LNCS-4103, pp.169–180, Springer, 2006, http://link.springer.com/chapter/10.1007/11837862_18, http://www.win.tue.nl/declare/wp-content/papercite-data/pdf/declarative_approach.pdf.
- [17] W.M.P. van der Aalst, M. Pesic, and H. Schonenberg, “Declarative workflows: Balancing between flexibility and support,” Computer Science - Research and Development, vol.23, no.2, pp.99–113, 2009, <http://link.springer.com/article/10.1007/s00450-009-0057-9> (Open Access), http://www.win.tue.nl/declare/wp-content/papercite-data/pdf/flexibility_support.pdf.
- [18] M. Pesic, H. Schonenberg, and W.M.P. van der Aalst, “DECLARE Demo: A Constraint-based Workflow Management System,” Proceedings of the BPM 2009 Demonstration Track, (ed.) B.W. A.K. Alves de Medeiros, 2009, <http://ceur-ws.org/Vol-489/paper1.pdf>.
- [19] M. Pesic, H. Schonenberg, and W. Aalst, “Declarative Workflow,” Modern Business Process Automation – YAWL and its Support Environment, (eds.) A.H.M. Hofstede, W.M.P. Aalst, M. Adams, and N. Russell, pp.175–201, Springer, 2010, <http://www.springer.com/br/book/9783642031205>, http://www.win.tue.nl/declare/wp-content/papercite-data/pdf/yawlbook_declarative.pdf.
- [20] F. Maggi, A. Mooij, and W. van der Aalst, “User-Guided Discovery of Declarative Process Models,” Symposium on Computational Intelligence and Data Mining, CIDM, pp.192–199, IEEE, 2011, <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5949297>, http://www.processmining.org/_media/publications/cidm2011.pdf.
- [21] J. Carmona, J. Cortadella, and M. Kishinevsky, “Genet: A Tool for the Synthesis and Mining of Petri Nets,” International Conferences on Application of Concurrency to System Design, ACS D, pp.181–185, 2009, <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5291046>.
- [22] M. Solé, and J. Carmona, “Rbminer: A Tool for Discovering Petri Nets from Transition Systems,” Automated Technology for Verification and Analysis: 8th ATVA 2010, LNCS-6252, pp.396–402, 2010, http://link.springer.com/chapter/10.1007/978-3-642-15643-4_33.
- [23] B.F. Dongen, A.K.A. Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. Aalst, “The prom framework: A new era in process mining tool support,” Applications and Theory of Petri Nets 2005: 26th ICATPN 2005, (eds.) G. Ciardo, and P. Darondeau, LNCS-3536, pp.444–454, Springer, 2005, http://link.springer.com/chapter/10.1007/11494744_25, <http://wwis.win.tue.nl/~wvdaalst/publications/p264.pdf>.
- [24] W.M.P. Aalst, B.F. Dongen, C.W. Günther, R.S. Mans, A.K.A. Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, and A.J.M.M. Weijters, “Prom 4.0: Comprehensive support for real process analysis,” Petri Nets and Other Models of Concurrency : 28th ICATPN 2007, (eds.) J. Kleijn, and A. Yakovlev, LNCS-4546, pp.484–494, Springer, 2007, http://link.springer.com/chapter/10.1007/3-540-44895-0_26, <http://alexandria.tue.nl/openaccess/Metis205749.pdf>.
- [25] W. van der Aalst, B. van Dongen, C. Gunther, A. Rozinat, H. Verbeek, and A. Weijters, “Prom: The process mining toolkit,” Proceedings of the BPM 2009 Demonstration Track, (ed.) B.W. A.K. Alves de Medeiros, 2009, <http://ceur-ws.org/Vol-489/paper3.pdf>.
- [26] H. Verbeek, J. Buijs, B. van Dongen, and W. van der Aalst, “Prom 6: The process

- mining toolkit,” Business Process Management Demonstration Track 2010, (ed.) M.L. Rosa, 2010, <http://ceur-ws.org/Vol-615/paper13.pdf>.
- [27] J. Claes, and G. Poels, “Process Mining and the ProM Framework: An Exploratory Survey,” Business Process Management Workshops: BPM 2012, (eds.) M. Rosa, and P. Soffer, LNBIP-132, pp.187–198, Springer, 2013, http://link.springer.com/chapter/10.1007/978-3-642-36285-9_19, <http://www.janclaes.info/post.php?post=pubbpi2012>.
- [28] W. van der Aalst, “Process Mining: A historical perspective,” Process Mining Camp 2013 基調講演, 2013, <http://fluxicon.com/camp/2013/slides/wil.pdf>.
- [29] C.W. Gunther, and B. Rozinat, “Disco: Discover Your Processes,” Proceedings of the Demonstration Track of the 10th Business Process Management (BPM 2012), (eds.) N. Lohmann, and S. Moser, 2012, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.416.9550&rep=rep1&type=pdf#page=46>.
- [30] B.F.V. Dongen, and W.M.P.V.D. Aalst, “A Meta Model for Process Mining Data,” EMOI Workshop - INTEROP’05 : Enterprise Modelling and Ontologies for Interoperability, (eds.) J. Casto, and E. Teniente, pp.309–320, 2005, <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-160/paper11.pdf>, <http://wwwis.win.tue.nl/~wvdaalst/publications/p261.pdf>.
- [31] B. van Dongen, “MXML A Meta model for process mining data,” , 2005, http://www.processmining.org/_media/presentations/miningmetamodelimoa2005.ppt.
- [32] H.M.W. Verbeek, J.C.A.M. Buijs, B.F. Dongen, and W.M.P. Aalst, “Xes, xesame, and prom 6,” Information Systems Evolution: CAiSE Forum 2010, (eds.) P. Soffer, and E. Proper, LNBIP-72, pp.60–75, Springer, 2011, http://link.springer.com/chapter/10.1007/978-3-642-17722-4_5, <https://pure.tue.nl/ws/files/3308208/Metis248868.pdf>.
- [33] W. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer, 2011, <http://www.springer.com/jp/book/9783642193446>, <http://www.processmining.org/book/start>.
- [34] A.V. Ratzer, L. Wells, H.M. Lassen, M. Laursen, J.F. Qvortrup, M.S. Stissing, M. Westergaard, S. Christensen, and K. Jensen, “CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets,” Applications and Theory of Petri Nets 2003: 24th ICATPN 2003, (eds.) W.M.P. Aalst, and E. Best, LNCS-2679, pp.450–462, Springer, 2003, http://link.springer.com/chapter/10.1007/3-540-44919-1_28.
- [35] A.K.A. de Medeiros, and C.W. Günther, “Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms,” Proceedings of the Sixth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools (CPN 2005), (ed.) K. Jensen, pp.177–190, University of Aarhus, 2005, <http://www.daimi.au.dk/CPnets/workshop05/cpn/papers/AnaKarlaAlvesdeMedeiros.pdf>, http://tmpmining.win.tue.nl/_media/publications/medeiros2005b.pdf.
- [36] A.K.A. de Medeiros, and C.W. Günther, “Process mining: Using cpn tools to create test logs for mining algorithms,” , 2005, http://www.processmining.org/_media/tools/cpn05_cpnlogs_v2.pdf.
- [37] G. Bergmann, A. Horvath, I. Rath, D. Varrato, “A benchmark evaluation of incremental pattern matching in graph transformation,” ICGT 2008, (eds.) H. Ehrig, R. Heckel, G. Rozenberg, and G. Taentzer, LNCS-5214, pp.396–410, Springer, 2008, http://link.springer.com/chapter/10.1007/978-3-540-87405-8_27, http://home.mit.bme.hu/~ahorvath/papers/BHRV_icgt08.pdf.

- [38] K. Hee, and Z. Liu, “Generating benchmarks by random stepwise refinement of petri nets,” PETRI NETS 2010, (eds.) S. Donatelli, J. Kleijn, R. Machado, and J. Fernandes, pp.403-417, CEUR-ws.org, 2012, http://ceur-ws.org/Vol-827/31_KeesHee_article.pdf.
- [39] A. Burattin, and A. Sperduti, “PLG: A Framework for the Generation of Business Process Models and Their Execution Logs,” Business Process Management Workshops: BPM 2010 International Workshops and Education Track, (eds.) M. zur Muehlen, and J. Su, LNBIP-66, pp.214-219, Springer, 2011, http://link.springer.com/chapter/10.1007/978-3-642-20511-8_20, <http://andrea.burattin.net/public-files/publications/2010-bpi.pdf>.
- [40] A. Burattin, “PLG2: Multiperspective Processes Randomization and Simulation for Online and Offline Settings,” arXiv.org, 2015, <http://arxiv.org/abs/1506.08415>, <http://arxiv.org/pdf/1506.08415v1.pdf>.
- [41] C. Ciccio, M.L. Bernardi, M. Cimitile, and F.M. Maggi, “Generating Event Logs Through the Simulation of Declare Models,” Enterprise and Organizational Modeling and Simulation: 11th International Workshop, EOMAS 2015, (eds.) J. Barjis, R. Pergl, and E. Babkin, LNBIP-231, pp.20-36, 2015, http://link.springer.com/chapter/10.1007/978-3-319-24626-0_2.
- [42] D. Calvanese, M. Montali, A. Syamsiyah, and W.M.P. van der Aalst, “Ontology-Driven Extraction of Event Logs from Relational Databases,” the 11th Int. Workshop on Business Process Intelligence (BPI 2015), LNBIP-, Springer, 2015, <https://www.inf.unibz.it/~calvanese/papers/calv-mont-syam-aals-BPI-2015.pdf>, <https://www.inf.unibz.it/~calvanese/papers-html/BPI-2015.html>.
- [43] S. Goedertier, D. Martens, J. Vanthienen, and B. Baesens, “Robust Process Discovery with Artificial Negative Events,” Journal of Machine Learning Research, Vol.10, pp.1305-1340, 2009, <http://www.jmlr.org/papers/volume10/goedertier09a/goedertier09a.pdf>, <http://jmlr.org/papers/volume10/goedertier09a/source/goedertier09a.tex>.
- [44] H. Ferreira, and D.R. Ferreira, “An integrated life cycle for workflow management based on learning and planning,” International Journal of Cooperative Information Systems, Vol.15, Issue.4, pp.485-505, 2006, <http://ejournals.wspc.com.sg/ijcis/ijcis.shtml>, <https://pdfs.semanticscholar.org/23f2/797db3c071e419cea20f665754621a7afabd.pdf>.
- [45] E. Lamma, P. Mello, F. Riguzzi, and S. Storari, “Applying Inductive Logic Programming to Process Mining,” Inductive Logic Programming, (eds.) H. Blockeel, J. Ramon, J. Shavlik, and P. Tadepalli, LNCS-4894, pp.132-146, Springer, 2008, http://link.springer.com/chapter/10.1007/978-3-540-78469-2_16, http://research.cs.wisc.edu/machine-learning/shavlik-group/ilp07wip/ilp07_lamma.pdf.
- [46] F. Chesani, E. Lamma, P. Mello, M. Montali, F. Riguzzi, and S. Storari, “Exploiting Inductive Logic Programming Techniques for Declarative Process Mining,” Transactions on Petri Nets and Other Models of Concurrency II: Special Issue on Concurrency in Process-Aware Information Systems, (eds.) K. Jensen, and W.M.P. van der Aalst, LNCS-5460, pp.278-295, Springer, 2009, http://link.springer.com/chapter/10.1007/978-3-642-00899-3_16, .
- [47] S. vanden Broucke, J. De Weerd, J. Vanthienen, and B. Baesens, “An improved process event log artificial negative event generator,” Technical Report FEB Research Report KBI-1216, Leuven (Belgium): KU Leuven Faculty of Economics and Business, 2012,

- http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2165204.
- [48] W. van der Aalst, P. de Crom, R. Goverde, K. van Hee, W. Hofman, H. Reijers, and R. van der Toorn, “ExSpect 6.4: An Executable Specification Tool for Hierarchical Colored Petri Nets,” *Application and Theory of Petri Nets 2000*, (eds.) M. Nielsen, and D. Simpson, LNCS-1825, pp.455–464, Springer, 2000, http://link.springer.com/chapter/10.1007/3-540-44988-4_26.
- [49] A. de Medeiros, B. van Dongen, W. van der Aalst, and A. Weijters, “Process Mining: Extending the Alpha-Algorithm to Mine Short Loops,” *BETA Working Paper*, WP 113, 2004, <http://wwis.win.tue.nl/~wvdaalst/publications/p221.pdf>.
- [50] W. van der Aalst, A. Weijters, and L. Maruster, “Workflow Mining: Discovering Process Models from Event Logs,” *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.9, pp.1128-1142, 2004, <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1316839>, <http://wwis.win.tue.nl/~wvdaalst/publications/p245.pdf>.
- [51] A. Weijters, and W. van der Aalst, “Process Mining: Discovering Workflow Models from Event-Based Data,” the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), (eds.) B. Kröse, M. de Rijke, G. Schreiber, and M. van Someren, pp.283–290, 2001, <http://wwis.win.tue.nl/~wvdaalst/publications/p128.pdf>.
- [52] C.J. Turner, A. Tiwari, R. Olaiya, and Y. Xu, “Process mining: from theory to practice,” *Business Process Management Journal*, vol.18, no.3, pp.493–512, 2012, <http://www.emeraldinsight.com/doi/abs/10.1108/14637151211232669>, https://dspace.lib.cranfield.ac.uk/bitstream/1826/7817/1/Process_Mining-theory_to_practice-2012.pdf.
- [53] A. Tiwari, C.J. Turner, and B. Majeed, “A review of business process mining: state-of-the-art and future trends,” *Business Process Management Journal*, vol.14, no.1, pp.5–22, 2008, <http://www.emeraldinsight.com/doi/full/10.1108/14637150810849373>, https://www.researchgate.net/publication/235300506_A_review_of_business_process_mining_State-of-the-art_and_future_trends.
- [54] C.D. Ciccio, and M. Mecella, “MINERful, a Mining Algorithm for Declarative Process Constraints in MailOfMine,” *Department of Computer and System Sciences Antonio Ruberti Technical Reports n. 3*, Sapienza University, 2012, http://ojs.uniroma1.it/index.php/DIS_TechnicalReports/article/view/9656, <ftp://ftp.repec.org/opt/ReDIF/RePEc/aeg/wpaper/2012-03.pdf>.
- [55] C.D. Ciccio, M.H.M. Schouten, M. de Leoni, and J. Mendling, “Declarative Process Discovery with MINERful in ProM,” *BPM Demo Session 2015*, (eds.) F. Daniel, and S. Zuga, pp.60–64, 2015, <http://ceur-ws.org/Vol-1418/paper13.pdf>.
- [56] E. Lamma, P. Mello, M. Montali, F. Riguzzi, and S. Storari, “Inducing declarative logic-based models from labeled traces,” *5th Business Process Management, BPM 2007*, (eds.) G. Alonso, P. Dadam, and M. Rosemann, LNCS-4714, pp.344-359, Springer, 2007, http://link.springer.com/chapter/10.1007%2F978-3-540-75183-0_25, .
- [57] E.M. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, and A. Sangiovanni-Vincentelli, “SIS: A System for Sequential Circuit Synthesis,” *Technical Report CA 94720*, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 1992, <https://www.eecs.berkeley.edu/Pubs/TechRpts/1992/2010.html>, <http://www.eecs.berkeley.edu/Pubs/TechRpts/1992/ERL-92-41.pdf>.

- [58] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Transactions on Information and Systems*, E80-D, no.3, pp.315–325, 1997, .
- [59] W. van der Aalst, A. Adriansyah, de Medeiros, et al., "Process Mining Manifesto," *BPM 2011 Workshops(Part I)*, (eds.) F. Daniel, K. Barkaoui, and S. Dustdar, LNBIP-99, pp.169-194, Springer, 2012, The original version: <http://www.springer.com/jp/book/9783642281143>.
- [60] W. van der Aalst, A. Adriansyah, de Medeiros, et al., "Process Mining Manifesto," , 2012, The final version: http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_manifesto.
- [61] W. van der Aalst, A. Adriansyah, de Medeiros, et al., "プロセスマイニングマニフェスト (最終版)," , 2012, <http://www.win.tue.nl/ieeetfpm/lib/exe/fetch.php?media=shared:pmm-japanese-v1.pdf>.

著者略歴

- [1] 飯島 正 (いいじま ただし)
慶應義塾大学 理工学部 専任講師 (管理工学科 所属). 慶應義塾大学 理工学部 計測工学科卒業 (1986年), 同大学院 理工学研究科 修士課程 修了 (1988年), 同博士課程 単位取得退学 (1991年). 1990年より (株) 東芝勤務を経て, 1992年より 現所属 (助手を経て現職). 博士 (工学). 情報システム学会 元理事 (2007–2008年 大会担当理事, 2009–2013年 理事).
- [2] 田端 啓一 (たばた けいいち)
2012年早稲田大学 大学院 修士課程修了, 同年, 日本電信電話 (株) に入社. 以来, ソフトウェアイノベーションセンタにて ソフトウェア工学の研究に従事. 専門分野: プログラム自動並列化, コンピュータアーキテクチャ, ソフトウェアテスト.
- [3] 斎藤 忍 (さいとう しのぶ)
2001年 慶應義塾大学 大学院 修士課程修了, 同年, (株) NTT データに入社. 2015年に日本電信電話 (株) に転籍. 現在はソフトウェアイノベーションセンタに所属. 上流プロセスに関する研究開発・技術支援に従事. 2007年 慶應義塾大学 大学院博士課程修了. 博士 (工学). 専門分野: ソフトウェア工学, 要求工学.