

[解説]

プロセスマイニング・サーベイ (第05回: 事例)

A Survey on Business Process Mining — 05: Case Studies —

飯島 正[†], 田端 啓一[‡], 斎藤 忍[‡]

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

[†]慶應義塾大学・理工学部

[‡]日本電信電話株式会社

[†]Faculty of Science and Technology, Keio University

[‡]Nippon Telegraph and Telephone Corporation

1 はじめに

プロセスマイニングにおけるプロセスには、形式的なものとそうでないものが混在しており、その両方を1つの枠組みで扱えるという利点がある。

研究の初期には、ログに先立って、そのログの元となる形式的なプロセスが存在することが前提とされた(図1)。このアプリオリ (a priori) なプロセスをマイニングするための代表的なアルゴリズムが、当連載解説の前回(第04回:アルゴリズム(1))で紹介した α -アルゴリズムである。利用例としては、プログラムから出力されたログを元に、プログラムの構造が持つプロセスを再発見することが挙げられる(図1)。

その後の研究で、ログの中から初めてプロセスが見つげ出されるような用途に、プロセスマイニングが応用され始めた(図2)。このように発見的なプロセスをマイニングする代表的なアルゴリズムが、次回(第06回:アルゴリズム(2))で紹介するヒューリスティック・マイナー (Heuristic Miner) である。利用例としては、人間の行動を記録したログを元に、人間の行動をプロセスとしてモデル化するような場合が当てはまる。

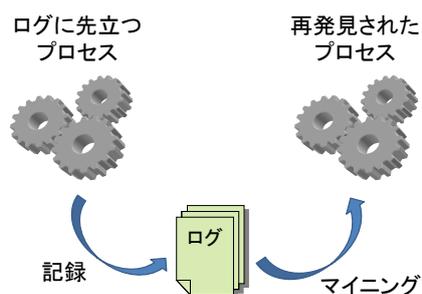


図1: 古くからある研究: ログに先立ってプロセスが存在する場合

一方で、プロセスマイニングマニフェスト^{[1][2]}では両者の区別を行わず、プロセスの「発見」「監視」「改善」が基本的な活動であるとして定義している。そこで、多岐に渡るプロセスマイニングの応用分野の中から、本稿では「発見」「監視」「改善」のそれぞれにおいて代表的な論文について述べる。

[解説] 2017年12月28日受付.

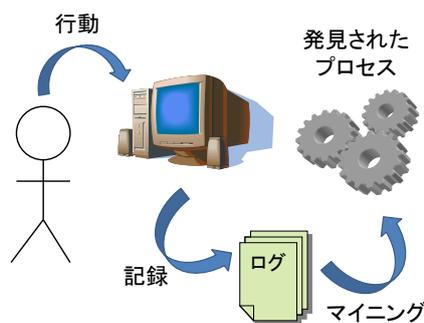


図 2: 最近の研究: ログの中から初めてプロセスが見つげ出される場合

2 発見・医療

2.1 概要

プロセスマイニングを応用した研究の中でも、取り分け大規模な「発見」の評価が行われたのが、Mansらによる文献^[3]である。この論文では、プロセスマイニングの従来手法であるヒューリスティック・マイナーやファジー・マイナー (Fuzzy Miner) といったアルゴリズムが、医療分野に適用可能であることの、初期段階の報告を行っている。

なお、医療においては、ログに先立って存在する明確なプロセスが存在しないため、ログの中から初めてプロセスを見つけ出す場合に該当する。

2.2 ログ

Mansらは、婦人腫瘍科を受診した患者 627 名分の 2 年分の診療情報をマイニング対象とした。婦人腫瘍科の患者には、他の診療科、例えば放射線科など、複数の診療科に渡って診察を受けるという特徴がある。医療現場には診療科ごとに数多くのコンピュータシステムが分散して存在するため、マイニングの対象となり得るログはあっても、散在するログどうしの関連付け作業は困難を伴う。この問題を解決するため、Mansらは、一切の医療行為が反映される病院の会計システムに着目し、そのログを利用した。このように、このケーススタディでは、プロセスマイニングを応用するにあたって、複数あるログのうちどれを利用するかという選択が重要であった(図 3)。

得られたログは、従来のプロセスマイニングが想定しているイベントの開始・終了の分離が行われていなかった。さらに、タイムスタンプの分解能が 1 日単位であり、同じ日の中の医療行為の順番はわからなかった。このようなログであっても、従来のプロセスマイニングの手法が適用できるのかが課題であった。

2.3 前処理

Mansらが得た会計システムのログには、患者に対する検査の詳細な項目、例えば、血中の個々の成分の検出のような、実験室レベルの粒度のイベントが含まれており、イベントの種類は 376 にのぼった。しかしながら、Mansらが欲しかったイベントは、診療科で発生する粒度のものであった。そこで、Mansらは、詳細な検査項目を、ログデータの前処理によって単に「検査」のように丸めこむことでイベントの種類を減らした。このように、プロセスマイニングの応用においては、Miner アルゴリズムを適用する以前の処理が重要である。

さらに、Mansらは、教師なし学習の一種である SOM(Self Organizing Map: 自己組織化写像) を用いてログをクラスタリングし、最大のクラスタから、類似する属性を持った 352 のケース (イベントト

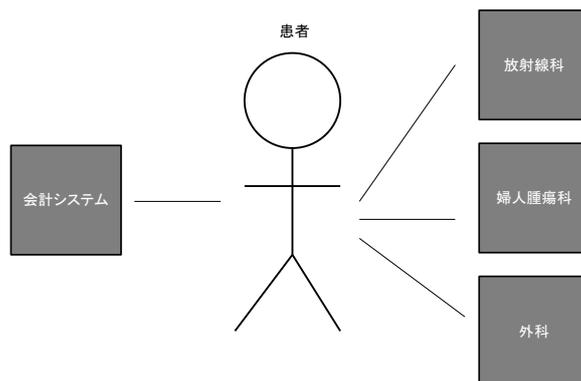


図 3: 医療における人とシステムの関わり

レース)を抽出した。

2.4 発見

SOMを適用した後のログから得られたヒューリスティック・マイナーの出力は、可読性の高いものであった(図4)。さらに、ファジー・マイナーも適用可能であった。なお、得られたプロセスモデルの評価は、病院の職員が正しさを確認するという定性的な評価にとどまっており、Mansらもまだ初期の取り組みであると述べている。しかしながら、特定ドメインにおける実データを用いてプロセスモデルを発見したケーススタディとして、本論文の貢献は大きい。

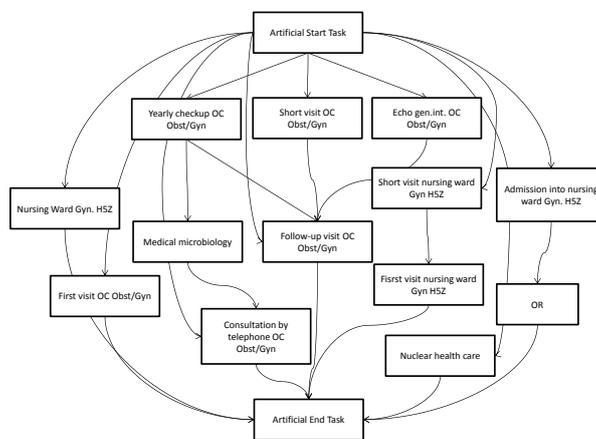


図 4: 医療におけるマイニング結果 [3]

2.5 所感

α -アルゴリズムに代表される従来のプロセスマイニングの研究では、ログに先立つアприオリなプロセスモデルを考え、これを再発見することを目指してきた。一方、Mansらが扱った医療においては、明確なプロセスモデルが存在せず、この点は大きな進歩である。

また、SOMを用いてデータをクラスタリングしている点は、今後、プロセスマイニングを応用していく上で、ログの前処理に機械学習やパターン認識を積極的に利用可能であることを示しているのではな

いか，ということを示唆している。

3 監視・ソフトウェア開発

3.1 概要

次に紹介する Rubin らによる参考文献^[4]は，マイニングされたプロセスから実際のログを「監視」するところまで実践を進めている。Rubin らは，オープンソース・ソフトウェア (OSS; Open Source Software) である ArgoUML のプロジェクトのソフトウェア構成管理 (SCM; Software Configuration Management) のログを解析し，プロセスモデルを発見するとともに，プロセスモデルに条件を与えることでログの適合性検証 (Comformance analysis) を実施している。

OSS にも様々なスタイルがあるが，一般的には，中央に版管理システム (VCS; Version Control System) のリポジトリが存在する。そして，VCS に変更を加える権限を持った開発者はコミッタと呼ばれる。その他の開発者やパッチ提供者は，コミッタに対してソースコードの差分を示し，採用を促す。

このような OSS での開発の場合も，ログに先立つ明示的なプロセスがあるわけではない。従って，ログの中からプロセスを初めて見つける必要がある。

3.2 ログ

解析の対象となる VCS は subversion である。Rubin らは，単純に“svn log”コマンドの内容をログとして取得した。“svn log”コマンドの出力例を図 5 に示す。図 5 の通り，個々のコミットのタイムスタンプや，コミットにおいて変更されたファイルのパスを取得できることがわかる。ただし，これだけではコミット以外にイベントと呼べるものが存在しないことになるため，Rubin らは前処理で工夫を行っている。

```
r210692 | rsandifo | 2014-05-21 20:00:35 +0900 | 44 lines
Changed paths:
 M /trunk/gcc/ChangeLog
 M /trunk/gcc/c/ChangeLog
 M /trunk/gcc/c/c-array-notation.c
 M /trunk/gcc/c-family/ChangeLog
 M /trunk/gcc/c-family/c-common.c
 M /trunk/gcc/c-family/c-common.h
 M /trunk/gcc/c-family/c-pretty-print.c
 M /trunk/gcc/c-family/c-ubsan.c
 M /trunk/gcc/c-family/cilk.c
 M /trunk/gcc/cp/ChangeLog
 M /trunk/gcc/cp/cp-array-notation.c
 M /trunk/gcc/cp/cvt.c
 M /trunk/gcc/cp/cxx-pretty-print.c
 M /trunk/gcc/cp/decl.c
 M /trunk/gcc/cp/error.c
```

図 5: ソフト開発におけるログ (gcc での例)

3.3 前処理

Rubin らは，前処理の過程で，“svn log”コマンドの出力から変更されたファイルのパスに着目し，ソースコードのディレクトリであれば“SRC”，テストコードのディレクトリであれば“TESTS”のような分類を行った。この単純化により，“SRC”，“TESTS”といった分類をイベントとし，従来のプロセスマイニングの手法を容易に適用可能とした。

3.4 発見

Rubin らはヒューリスティック・マイナーを適用したあと、目的のプロセスモデルをペトリネットとして「発見」した(図 6)。Rubin らはさらに、ペトリネットに色をつけることで、頻度の高いトランジションを明確化し、プロセスモデルを拡張している。

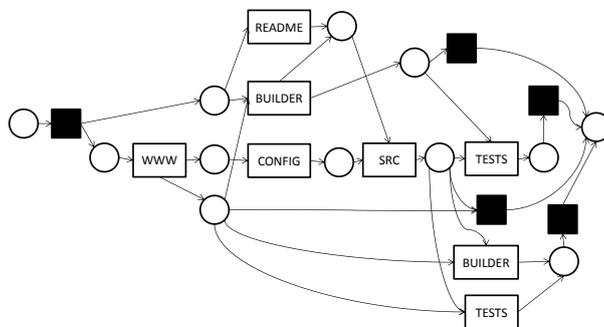


図 6: ソフト開発におけるマイニング結果 [4]

3.5 監視

得られたペトリネットについて、Rubin らは、「SRC と TESTS が別な開発者に実行される必要がある」というモデルへ拡張を行い(図 7)、VCS のログに対して実際に適合性検証を実施した。その結果、SRC と TESTS が同じ開発者によって実行されている箇所を特定することができた。

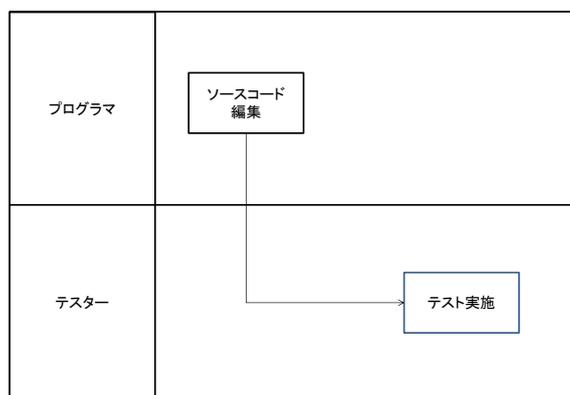


図 7: ソフト開発におけるルールの導入

3.6 所感

Rubin らの取り組みのうち、「発見」の部分は、MSR(Mining Software Repositories)の一環と考えることができる。また、従来のプロセスマイニング手法を適用するにあたっての前処理手法は、本論文の大きな貢献であると筆者は考える。「監視」の部分は、OSS のユニットテストは開発者が自分で作成する

のが一般的ではないかという疑問は残るものの、実際のログに対して適合性検証まで行っている点で進んでいる。

4 改善・半導体製造業

4.1 概要

最後に紹介する研究は、半導体製造装置のテストプログラムが出力するログをマイニングし、得られたプロセスを「強化」につなげた、参考文献^[5]である。

ASML社は半導体製造装置の製造会社である。半導体製造装置は非常に精密な光学機器であるため、稼働体制に入る前に、繊細な調整を伴うテストを繰り返す必要がある。ASML社の場合、半導体製造装置に対して、出荷前、ならびに客先での計2回、同一のテストを数週間に渡って実施する。半導体の微細化は日進月歩であり、顧客は最先端の装置を欲するため、Time-to-Marketは重要な課題である。そこで、この数週間に渡るテストを短縮してTime-to-Marketを改善したいというのがRozinatらのモチベーションである。

なお、半導体製造装置のテストはプログラムによって行われるため、ログに先立ってプロセスが存在する場合であると言える。

4.2 ログ

Rozinatらは、出荷前のテストで出力されたログをマイニングし、プロセスモデルを作成した。ログはそれぞれタイムスタンプとアルファベット4文字のコードから成る(図8)。Rozinatらはこの4文字のコードをイベントとみなした。

```
1596, 31-01-2006 17:33:13, 31-01-2006 17:33:39, POLA
1596, 31-01-2006 17:33:50, 31-01-2006 17:34:46, OSWL
1596, 31-01-2006 17:34:48, 31-01-2006 17:35:10, OSSP
1596, 31-01-2006 17:36:18, 31-01-2006 17:36:49, AHZI
1596, 31-01-2006 17:42:18, 31-01-2006 17:43:25, DSNA
1596, 31-01-2006 17:43:39, 31-01-2006 17:44:56, AHZI
1596, 31-01-2006 17:44:57, 31-01-2006 17:59:10, SVEI
1596, 01-02-2006 07:15:37, 01-02-2006 07:33:25, SVEI
1596, 01-02-2006 07:35:00, 01-02-2006 07:53:24, SCEI
1596, 01-02-2006 07:53:25, 01-02-2006 07:54:58, YHLH
1596, 01-02-2006 07:54:59, 01-02-2006 07:57:41, AHHJ
1596, 01-02-2006 07:57:42, 01-02-2006 08:04:40, AHCA
```

図 8: 半導体製造業におけるログ^[5]

4.3 前処理

初め、360種類あったイベントは、複数のテストで共通に出力されるものを除外することで、テストの種類と対応する70種類のイベントに削減された。

4.4 発見

「発見」にはHeuristic Minerが適用され、可読性の高いプロセスモデルが出力された。

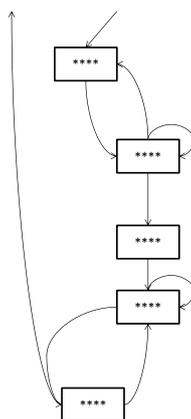


図 9: 半導体製造業におけるマイニング結果 (抜粋)^[5]

4.5 強化

Rozinat らは、得られたテストのプロセスモデルに対して、パフォーマンスボトルネックの解析を行った。その結果、テストはフィードバックループから構成されており、このループが最も時間を要していることが判明した（「改善」）。図 9 はそのマイニング結果の抜粋で、ループを含んでいることがわかる。

4.6 所感

このケーススタディでは、4 文字のコードをユニークなテストイベントと紐付けてプロセスを発見する過程について主に述べられている。実際には、テストプログラムおよびテスト手順書からテストのプロセスモデルが得られると推測されるが、これを自動で「発見」することに価値がある。

5 その他の応用

現在、多くのグループがプロセスマイニングの研究を行っているが、中でも van der Aalst らのグループは、プロセスマイニング研究の初期からアルゴリズムを提案しており、その後も、応用の研究を精力的に行ってきた。そこで、本稿では、van der Aalst らの研究グループの論文に着目し、プロセスマイニングを各分野で実際に応用した最新の論文・出版物について、「発見」「監視」「改善」の 3 つの評価軸でまとめた (表 1)。「発見」を行っている論文は多いものの、「監視」「強化」を行っている論文は少ない。

分野	発見	監視	強化
Medical ^[3]	○	-	-
Social Network ^[6]	-	-	-
Business Data ^[7]	○	-	-
Security ^[8]	-	○	-
Ubiquitous ^[9]	○	-	-
Semiconductor ^[5]	○	-	○
Staff Assignment ^[10]	○	-	-
Software Development ^[4]	○	○	-
Multi-agent ^[11]	○	○	○

表 1: プロセスマイニングの応用

6 おわりに

本稿では、医療、ソフトウェア開発、半導体製造業といった、三つの応用分野で、「発見」、「監視」、「改善」といったプロセスマイニング技術が目的とする典型的な活動への適用事例を紹介した。

本連載解説では、数回にわたり技術調査の結果を示してきたが、具体的な適用事例が十分に示されていないため、プロセスマイニング技術がどのように活用されてきたか、これからどのような分野にどのような形で活用していくかが、わかりにくかった。そこで、プロセス発見アルゴリズムの調査報告の途中ではあるが、事例解説を行うこととした。今回の解説が、プロセスマイニング技術活用の裾野を広げる一助になることを期待している。

今回の執筆分担:

本稿の執筆分担は田端・斎藤が担当し、飯島が内容の確認と字句・文体の統一・調整を行った。

文献

- [1] 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, LNBI-99, pp.169-194, Springer, 2012, The original version: <http://www.springer.com/jp/book/9783642281143>.
- [2] 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.
- [3] R. Mans, M. Schonenberg, M. Song, W. van der Aalst, and P. Bakker, "Application of process mining in healthcare -a case study in a dutch hospital," in Biomedical Engineering Systems and Technologies, (eds.) A. Fred, J. Filipe, and H. Gamboa, 25 of Communications in Computer and Information Science, pp.425-438, Springer Berlin Heidelberg, 2009, https://link.springer.com/chapter/10.1007/978-3-540-92219-3_32,http://dx.doi.org/10.1007/978-3-540-92219-3_32.
- [4] V. Rubin, C. Günther, W. van der Aalst, E. Kindler, B. van Dongen, and W. Schäfer, "Process mining framework for software processes," in Software Process Dynamics and Agility, (eds.) Q. Wang, D. Pfahl, and D. Raffo, 4470 of Lecture Notes in Computer Science, pp.169-181, Springer Berlin Heidelberg, 2007, https://link.springer.com/chapter/10.1007/978-3-540-72426-1_15,http://dx.doi.org/10.1007/978-3-540-72426-1_15.
- [5] A. Rozinat, I. de Jong, C. Gunther, and W. van der Aalst, "Process mining applied to the test process of wafer scanners in asml," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 39, 4, pp.474-479, July 2009, <https://ieeexplore.ieee.org/document/4806028>,<http://dx.doi.org/10.1109/TSMCC.2009.2014169>.
- [6] W. van der Aalst, and M. Song, "Mining social networks: Uncovering interaction patterns in business processes," in Business Process Management, (eds.) J. Desel, B. Pernici, and M. Weske, 3080 of Lecture Notes in Computer Science, pp.244-260, Springer Berlin Heidelberg, 2004, https://link.springer.com/chapter/10.1007/978-3-540-25970-1_16,http://dx.doi.org/10.1007/978-3-540-25970-1_16.
- [7] S. Dustdar, T. Hoffmann, and W. van der Aalst, "Mining of ad-hoc business processes with teamlog," Data & Knowledge Engineering, 55, 2, pp.129 - 158, 2005, <http://www.sciencedirect.com/science/article/pii/S0169023X05000133>,<http://dx.doi.org/10.1016/j.datak.2005.02.002>.
- [8] W. van der Aalst, and A. de Medeiros, "Process mining and security: Detecting anomalous process executions and checking process conformance," Electronic Notes in Theoretical Com-

- puter Science, 121, pp.3 - 21, 2005, <http://www.sciencedirect.com/science/article/pii/S1571066105000228>, <http://dx.doi.org/10.1016/j.entcs.2004.10.013>.
- [9] A. de Medeiros, B. van Dongen, W. van der Aalst, and A. Weijters, "Process mining for ubiquitous mobile systems: An overview and a concrete algorithm," in Ubiquitous Mobile Information and Collaboration Systems, (eds.) L. Baresi, S. Dustdar, H. Gall, and M. Matera, 3272 of Lecture Notes in Computer Science, pp.151-165, Springer Berlin Heidelberg, 2005, https://link.springer.com/chapter/10.1007/978-3-540-30188-2_12, http://dx.doi.org/10.1007/978-3-540-30188-2_12.
- [10] S. Rinderle-Ma, and W.M. van der Aalst, "Life-cycle support for staff assignment rules in process-aware information systems," Technical Report 213, TU Eindhoven, 2007, <http://dbis.eprints.uni-ulm.de/373/>.
- [11] A. Rozinat, S. Zickler, M. Veloso, W. van der Aalst, and C. McMillen, "Analyzing multi-agent activity logs using process mining techniques," in Distributed Autonomous Robotic Systems 8, (eds.) H. Asama, H. Kurokawa, J. Ota, and K. Sekiyama, pp.251-260, Springer Berlin Heidelberg, 2009, https://link.springer.com/chapter/10.1007/978-3-642-00644-9_22, http://dx.doi.org/10.1007/978-3-642-00644-9_22.

著者略歴

[1] 飯島 正 (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 NTT データに入社. 2015 年 日本電信電話株式会社に転籍. 現在, ソフトウェアイノベーションセンタに所属. 2016 年よりカリフォルニア大学アーバイン校 客員研究員. ソフトウェア工学, 要求工学に関する研究開発に従事. 2007 年 慶應義塾大学 大学院 博士課程修了. 博士 (工学).