Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1199–1202
DOI: 10.15439/2017F372
ISSN 2300-5963 ACSIS, Vol. 11

Process Mining Methods for Post-Delivery Validation

Paweł Markowski, Michał R. Przybyłek
Polish-Japanese Academy of Information Technology
Warsaw, Poland
Email:{pawel.markowski, mrp}@pjatk.edu.pl

Abstract—**The aim of this paper is to show the strengths and the weakness of process mining tools in post-delivery validation. This is illustrated on two use-cases from a real-world system. We also indicate what type of research has to be done to make process mining tools more usable for validation purposes.**

I. Introduction

THE lean department of a real-world company asked us to check the control system of a production line against the expected cycle of manufacturing. This is the usual process of validation in lean manufacturing and software development [1], [2], [3] — since in such methodology there is no up-front design, one evaluates in the working environment whether the implemented system meets the expectations and needs of the principals. The lean department is responsible for production process optimization that leads to overall increase in efficiency. They focus on layout optimization and usability development to ensure best environment to work and high throughput. They decided to try new methods of material flow analysis by leveraging process mining features. Our evaluation is aligned with lean thinking adopted by the company.

We gathered the data from the warehouse management system and production line (presented in Table I) and used them to discover the real process that was followed by the mechanical parts manufactured on the production line. Our main tool in process-discovery [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15] was an open source platform ProM [16] with various plugins (i.e. Directly-follows Graph, IVM, Discover Graph). After examination of the discovered processes by the principals, serious anomalies were discovered, which led to the reimplementation of the system. However, not all of the processes were actually discovered *correctly*. Despite feeding them with information about the duration of each action in the process, the mining algorithms were unable to discover correctly the parallelism of the actions. Consequently, they produced large clumsy and meaningless diagrams. This shows the limitations and weakness of the currently available methods.

The paper is organized as follows. The next section describes the gathered data and the platform that was used to mine processes. In Section III we describe the use-case of warehouse moves and the corollaries of our analysis. In Section IV we deal with a use-case containing parallel actions and show that contemporary algorithms fail to mine a useful model. The paper is concluded in Section VI, where we also suggest some further research in the area of process discovery.

II. Discovering processes

Discovering processes from event logs requires a collection of events with timestamps and case ID, which identifies instance of executed process. Timestamps allow process mining algorithms to transform the data into diagrams, which represent discovered models. According to a given set of parameters model accuracy can be different. By leveraging more features it is possible to i.e. receive a less accurate graph, which may be easier to analyze or interpret. For the validation of the system we used ProM software (version 6.6). We split tests into two parts — the first one is the classic process discovery with events triggered sequentially. In the second case, some of the actions were executed in parallel. It requires proper approach, which will be able to identify specific flows with parallel actions.

There are many methods and forms of storing event logs of information systems. Each solution may have own approach how to collect and store event logs. When an event occurs, the system generates a set of data about the action that triggered the event. Information included in it can be stored in a specified location, like raw file or a database record. There are often special rules that indicate, which information should be stored in the log. Many systems have different levels of log detail, which can be setup during configuration. To start working with process mining tool ProM, we have to deliver an unified log file. ProM allows conversion from CSV to XES format — an XML uniform format of data recognized by the platform. It has a dedicated creator module that allows a user to easily perform transformations. The greatest issue here is the quality of data stored in log files.

III. Case: Warehouse moves analysis

We focused on warehouse movement analysis. The system was modified to record each move performed in the warehouse area. It ensured better understanding of daily basis operations and, hopefully, will help in further optimization processes in the department. We collected event logs (shown in Table I) that describe actions with precise timestamps. Case ID reflects single pallet of goods.

A discovered model of the process, shown on Figure 1, has accuracy comparable to human expert knowledge about the

Table I
AN EVENT LOG GATHERED FROM THE WMS.

Case ID	Actor	Time Stamp	Event
218,833	328	2017-04-11 07:35:06	put_in
218,833	233	2017-04-23 22:57:13	qty_change
218,833	233	2017-04-23 22:57:13	put_out
219,897	328	2017-04-18 10:38:33	produced
219,897	328	2017-04-18 10:42:33	putting_in
219,897	328	2017-04-18 10:42:46	put_in
219,897	234	2017-04-27 00:05:50	qty_change
219,897	234	2017-04-27 00:05:50	put_out
217,128	230	2017-04-03 07:00:21	produced
217,128	328	2017-04-03 08:16:38	putting_in
217,128	328	2017-04-03 08:16:48	put_in
217,128	328	2017-04-03 11:11:04	qty_change
217,128	328	2017-04-03 11:11:04	put_out
220,006	229	2017-04-18 20:00:56	qstatus_1
220,006	229	2017-04-18 20:00:57	unload
220,006	161	2017-04-20 02:30:12	qstatus_2
220,006	420	2017-04-20 21:41:59	putting_in
220,006	420	2017-04-20 21:47:24	put_in
220,006	328	2017-04-22 11:28:01	qty_change
220,006	328	2017-04-22 11:28:01	put_out
219,7	229	2017-04-14 06:59:45	qstatus_1
219,7	229	2017-04-14 06:59:47	unload
219,7	161	2017-04-24 13:46:48	qstatus_6
219,7	161	2017-04-25 15:27:50	qstatus_2
219,7	321	2017-04-26 12:03:54	qty_change
219,7	321	2017-04-26 12:03:54	put_out
220,898	251	2017-04-22 08:01:28	qstatus_1
220,898	251	2017-04-22 08:01:28	unload
220,898	91	2017-04-22 09:12:53	qstatus_2
220,898	251	2017-04-22 13:14:43	putting_in
220,898	251	2017-04-22 13:14:48	put_in
220,898	321	2017-04-23 06:50:56	qty_change
220,898	321	2017-04-23 06:50:56	put_out
217,187	321	2017-04-03 09:48:42	qstatus_1
217,187	321	2017-04-03 09:48:42	unload
217,187	214	2017-04-04 12:48:15	qstatus_2
217,187	328	2017-04-04 12:49:52	qty_change
217,187	321	2017-04-04 13:07:56	qty_change
217,187	321	2017-04-05 09:23:09	putting_in
217,187	321	2017-04-05 09:25:54	put_in
217,187	321	2017-04-06 04:16:26	qty_change
...	...	...	...

real model of the process. The discovered model distinguishes two areas, which have different starting points.

The first one is a warehouse responsible for storing inbound components. Most of part numbers have additional quality control, which is performed by internal laboratories. Quality inspectors control incoming wares and change status after measurements. Prototypes and parts conforming to the standards and specifications are stored in warehouse racks and shelves. Production department order trigger move in warehouse that leads to release of a proper number of parts.

The area that is responsible for the shipments (outbounds) is described on Figure 1. It starts from "produced" action. Produced goods are stored in outbound warehouse. Goods can be put into specified rack or can be directly moved to carriers' truck. When the truck arrives, warehouse employees use terminal that gives them information about, which pallet have to be loaded into the track. The system enforces compliance with FIFO methodology.

Streamlining just in time production is one of the most

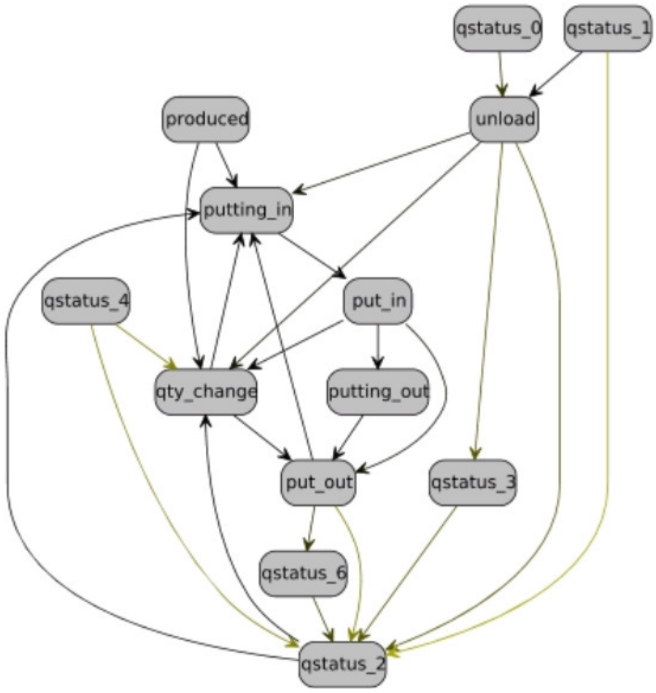

Figure 1. Process discovered by ProM Casual Activity Graph from event log presented in Table I.

valuable optimization for the company. Improvement will be apparently visible in key process indicators. Parts are stored in warehouse in a racks after production what is presented on Figure 2. Storing wares on shelves leads to freeze assets that could be used to gain competitive advantage. Figure 3 mined by Inductive Visual Miner presents goods flow in a warehouse. It gives a possibility to monitor moves on the animation with a time-line and filters. Presented activities can be tracked and verified. The company has to focus on production plans and on reorganization of the transports, which leads to downsizing time that stored staff spends on shelves. Modern process mining algorithms, implemented in ProM software, can perfectly reflect process model [17], [18], [19], where actions are not performed in parallel.

IV. CASE: PRODUCTION TRACEABILITY LOGS WITH PARALLEL EVENTS

Most of the production lines have specialized, dedicated software solution, which is responsible for collecting production events log. This kind of solution is required for most demanding and restrictive areas like pharmacy or automotive. This functionality gives an opportunity to recall from the market specified batch of defective items. Without traceability and its archive module company won't be able to specify, which item batches have to be removed from the market.

The production line has dedicated traceability database: Microsoft® SQL Server® 2008 R2 SP2 – Express Edition. In this paper we analyze a part of the line from the perspective of human $\leftrightarrow$ machine interaction, which is realized by parallel

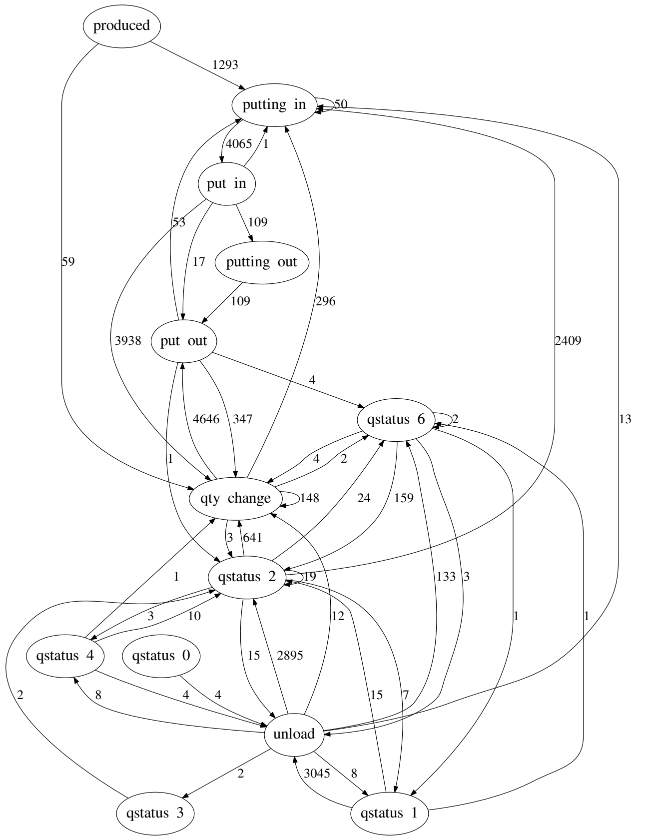

Figure 2. Process discovered by ProM Directly Follows Graph from event log presented in Table I.

quality inspection tasks. Operator activity records are stored only when a person rejects a given part during a specified operation.

An operator is a person responsible for supervising process performed on the station. Visual inspection of the components that are loaded to a machine and final product check are crucial and can minimize scrap costs. When operation is done, operator checks an item to minimize risk of defectiveness / incomplete produced item. If an item has been rejected earlier, the organization covers a lower cost of the scrap. The cost includes i.e. production time, components used during production process, energy, machine utilization, spare parts and so on. During quality validation operators can mark part as rejected (not valid) with an error code, which describes rejection reason.

Operations performed on a machine are recorded in a separated database table. This database contains identifiers of produced parts or its batches. Additionally, there are attached operation parameters and sensors values, which describe the results of each operation.

From the process mining perspective, there are interesting parallel activities between a machine and an operator involved in the production step.

Logs from each operation are stored in independent tables. To ensure proper model recognition consistent file log is required. Source log file has to be combined from extracted data tables. One of the tables contains information about visual inspection performed by human.

HMI panel is installed on the machine 605. This device allows operator to mark an item as rejected. It occurs when the item does not meet quality restrictions during visual inspection. During production process events together with their parameters are stored in the database archive. Operations OP605 and OP605HMI are parallel activities whose time of execution is the same.

In this case, timestamps with information when each singe item was completed on the stations are the same. Unfortunately the discovered process was flattened and its actions were shown as if they had been performed sequentially. Figure 4 represents graph, where operations OP605 and OP605 HMI are executed one after another. This flow is not align to the existing process implemented in the production process.

V. Parallel activities

For the validation, a subset of production line was checked by the algorithms implemented in ProM tool. The production line logs stored in the database have a various number of columns. These columns contain information about serial numbers, specified models, item status/error codes, cycle times and lot of parameters measured during a specific operation. We calculated the starting time of actions using the ending time of operations and the cycle time. Information about intervals was loaded to ProM XES file. The Start Time and Completion Time have been entered in PROM CSV to XES converter. The result of this operation was similar, because discovered process diagram had the same sequence. A single action was split into two separated actions (i.e. OP605 Start and OP605 Completed).

Results of the generated model (Figure 5) compared to the layout elaborated by a team of engineers do not reflect parallel operations performed at the set of machines.

VI. Conclusions and Further Work

This paper describes some aspects of a validation of the process of manufacturing mechanical parts on the production line, which is aligned with lean thinking adopted by a real-world company. We successfully applied available process mining algorithms to validate the production line and provided valuable suggestions to the lean department. On the other hand, the algorithms were of no use when it came to the discovery of highly parallel processes. Moreover, feeding the algorithms with additional knowledge about the duration of each action in the process did not help. This suggests that to make process mining tools more beneficial for the validation purposes, further research should focused on parallel process discovery (with and without additional information about duration of the actions).

Figure 3. Process discovered with filtered actions and paths by ProM Inductive Visual Miner from event log presented in Table I.

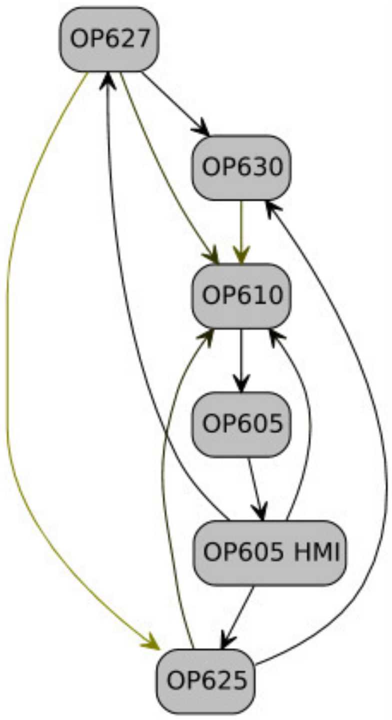

Figure 4. Process discovered by ProM Casual Activity Graph from event log.

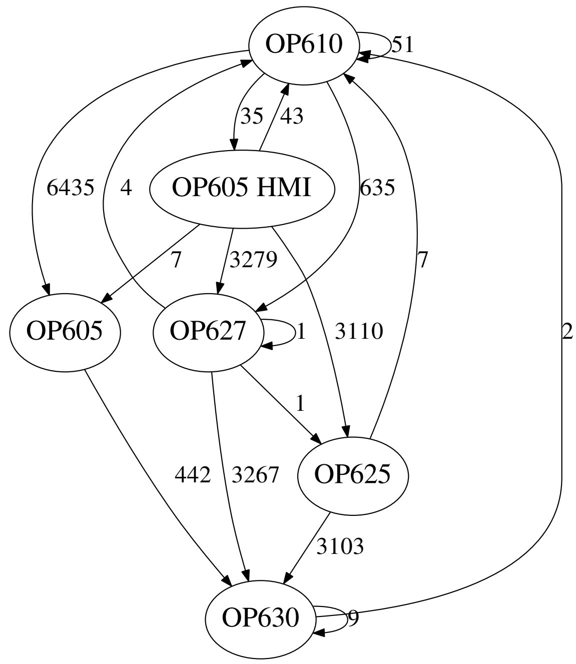

Figure 5. Process discovered by ProM Directly Follows Graph from event log.

References

[1] P. Rodriguez, J. Markkula, M. Oivo, K. Turula, *Survey on agile and lean usage in finish software industry*, ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Lund, Sweden, 2012, doi: 10.1145/2372251.2372275

[2] M. Poppendieck, T. Poppendieck, *Leading Lean Software Development: Results Are not the Point*, Addison-Wesley Signature Series, 2009, isbn: 9780321699657

[3] M. Poppendieck, T. Poppendieck, *Lean Software Development: An Agile Toolkit*, Addison-Wesley, 2013, isbn: 0321150783.

[4] W.M.P. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer Verlag, 2011.

[5] E.M. Gold, *Language identification in the limit*, Information and Control, Volume 10, 1967.

[6] D. Angluin, *Inductive Inference of Formal Languages from Positive Data*, Information and Control, Volume 42, 1980.

[7] L.G Valiant, *A theory of the learnable*, Communications of The ACM, volume 27, 1984.

[8] W.M.P. van der Aalst, B. van Dongen, *Discovering Workflow Performance Models from Timed Logs*, Engineering and Deployment of Cooperative Information Systems, pp. 107-110, 2002.

[9] L. Wen, J. Wang, J. Sun, *Detecting Implicit Dependencies Between Tasks from Event Logs*, Lecture Notes in Computer Science, Volume 3841, 591-603, 2006.

[10] C. Ren, L. Wen, J. Dong, H. Ding, W. Wang, M. Qiu, *A Novel Approach for Process Mining Based on Event Types*, IEEE SCC 2007, 721-722, 2007.

[11] A.K. Medeiros, A.J. Weijters, W.M.P. van der Aalst, *Genetic process mining: an experimental evaluation*, Data Mining and Knowledge Discovery, Volume 14 Issue 2, 2007.

[12] J.E. Cook, A.L. Woolf, *Discovering models of software processes from event-based data*, ACM Transactions on Software Engineering and Methodology, Volume 7 Issue 3, 1998.

[13] A. Brazma, *Efficient algorithm for learning simple regular expressions from noisy examples*, Workshop on Algorithmic Learning Theory ALT'94, Lecture Notes in AI, Volume 872, Springer, 1994.

[14] J. Herbst *A Machine Learning Approach to Workflow Management*, 11th European Conference on Machine Learning, Lecture Notes in Computer Science, Volume 1810, 2000.

[15] M.R. Przybyłek *Skeletal algorithms*, International Conference on Evolutionary Computation Theory and Applications 2011, pages 80-89

[16] *ProM — an extensible framework that supports a wide variety of process mining techniques*, http://www.promtools.org

[17] R. Mans, W.M.P. van der Aalst, R. Vanwersch *Process Mining in Healthcare*, Springer Briefs in Business Process Management; Springer International Publishing: Cham, Germany, 2015

[18] C. Fernandez-Llatas, A. Lizondo1, E. Monton, J-M Benedi, V. Traver *Process Mining Methodology for Health Process Tracking Using Real-Time Indoor Location Systems* Sensors, 11/2015; 15(12):29821-29840. DOI: 10.3390/s151229769

[19] P. Markowski, M.R. Przybyłek *Process Mining Methodology in Industrial Environment: Document Flow Analysis* Proceedings of the Federated Conference on Computer Science and Information Systems 2016, ACSIS, Vol. 8. ISSN 2300-5963, pp. 1175–1178, doi: 10.15439/2016F456