

Understanding Production Chain Business Process Using Process Mining: A Case Study in the Manufacturing Scenario

Alessandro Bettacchi^(✉), Alberto Polzonetti, and Barbara Re

Computer Science Division, University of Camerino, Camerino, Italy
{alessandro.bettacchi,alberto.polzonetti,barbara.re}@unicam.it

Abstract. Due to the continuous market change the enterprises need to react fast. To do that a better understanding of the way to work is needed. Indeed this was a real need of a manufacturing enterprise working in the production of coffee machines and selling them all over the world. In this paper, we present the experience made in the application of process mining techniques on a rich set of data that such enterprise collected during the last six years. We compare five mining algorithms, such as: α -algorithm, Heuristics Miner, Integer Linear Programming Miner, Inductive Miner, Evolutionary Tree Miner. We evaluated algorithms according to specific quality criteria: fitness, precision, generalization and simplicity. Even if comparison studies are already available in the literature we check them according to our working context. We conclude that the Inductive Miner algorithm is especially suited for discovering production chain processes in the context under study. The application of process mining gives the enterprise a comprehensive picture of the internal process organization. Resulting models were used by the company with successful results to motivate the discussion on the need of developing a flexible production chain.

Keywords: Process mining · Process discovery · Business process · ProM Framework · Mining algorithm · Production chain

1 Introduction

Manufacturing companies need to survive in a global market that asks for continuous align of the production and the internal organization, to the needs coming from the market. To do that a better understanding of the enterprise on the way to work is needed. This gives also the possibility to continuously improve production, avoid bottleneck and unwanted behaviors, or even workarounds enforced by the workers. This was the need of the manufacturing company working in the production of Coffee Machines motivated our work. After several meetings with the management board of such company we concluded that “a depth investigation of the production process is mandatory to learn from the past and to continuously improve the way to work”. The company wanted to check the

validity of the production processes, especially in exceptional events, for understanding optimizations to be done in order to save time and money and to have a more effective organization of work.

We based our study on such a real case study. The enterprise already have an internal traceability systems as part of its technology infrastructure. Based on the rich set of collected information (called log) via such system we run process mining techniques. Process mining aim is to extract non-trivial and useful information from event logs available in current information systems for discovering, monitoring and improving real processes [5]. Process mining is “evidence-based”, it ensures a close correspondence between modeled and observed behavior because the evaluation and definition of the model is based on the real performance of the process. Moreover Process mining is based on facts, the event data, and it is addressed to discover the current “as-is” processes [15]. We use 450.000 events concerning six years production of 32 different products. At the end of the study we give to the company a more comprehensive picture of the internal process organization. The use of mining techniques allows to evaluate which is the historical course of the manufacturing process. Such awareness support the manager to enhance the entire production process.

This paper reports the results of the application of process mining in the production chain. We assess and compare five mining algorithms such as α -algorithm, Heuristics Miner, Integer Linear Programming Miner, Inductive Miner, and Evolutionary Tree Miner. In particular, we compared the algorithms performances according to specific quality criteria: fitness, precision, generalization and simplicity. Even if this is not the first study aiming to compare such algorithms [3, 7], we run such analysis on our context to check general results on a specific application scenario.

The paper is organized as follow. Section 2 describes the case study, while Sect. 3 reports some background material. Section 4 gives an overview of the results coming from the application of the process mining algorithms on the case study. Finally, Sect. 5 reports conclusions and opportunities for further research.

2 Working Scenario

The case study refers to a manufacturing company producing and selling worldwide coffee machines since more than seventy years. The production chain relies on assembling components provided by several suppliers or internally produced. The production process is spread over six production lines enumerated from 1 to 6. Each production line, then, is organized into stations with specific objectives. The stations are identified by the letters A to F. According to the different types of coffee machines the organization of the stations in production lines can change. For production lines 5 and 6 there are only 5 stations: the station C is not present and all its activities are executed in station B.

The following are details of the stations.

- Station A starts with the activation of an RFID, which is used to uniquely identify a coffee machine. The second step is assembling the body and the

frame of a new coffee machine. The activated RFID is then associated to the new coffee machine.

- Station B handles the hydraulic system: boiler installation, pipes assembly, etc. Considering the production lines from 1 to 4 in Station B only a portion of the hydraulic system is assembled, while in production lines 5 and 6 the entire hydraulic system is assembled.
- Station C in the production lines from 1 to 4 completes the assembly of the hydraulic system. In addition, when required by a particular quality certification, a test of hydraulic circuit is also performed. In production lines 5 and 6 this station is not considered.
- Station D deals with electrical circuit: control unit, wiring, electrical connections between control unit and components are installed.
- Station E performs the testing. It is executed simultaneously for multiple coffee machines.
- Station F completes the coffee machine including in the packaging manuals, accessories and identification tag.

Production lines activities are also supported by two areas of pre-assembly in which some electrical components (i.e. push button panels and electric cards) are set up and some types of boiler are built.

From IT infrastructure point of view the enterprise uses several information systems. In particular, the production lines, and related stations, are managed by a customized Process-Aware Information System named Automatic System for Chain COntrol (ASCCO). ASCCO traces all the information related to the production line (assembly steps, assembly times, faults, repairs, ...). ASCCO also integrates warehouse management providing real-time localization of components and assembled coffee machines.

3 Process Mining Techniques

Process mining techniques are grouped into four classes: *Discovery*, *Conformance*, *Enhancement* and *Operational Support* [15]. We concentrate on *Discovery* and already defined mining algorithms [14]. This section provides an overview of mining algorithms and a presentation of the tool and methods we use to run our research.

During the past 15 years, various process discovery techniques have been proposed based on different approaches. In this work we focus on five of the most common algorithms: α -algorithm [17]; HeuristicsMiner (HM) [20]; Integer Linear Programming (ILP) Miner [18]; Inductive Miner (IM) [6, 7]; Evolutionary Tree Miner (ETM) [1]. These five algorithms were selected for two main reasons: their availability in ProM 6.5¹ [19], the tool used for process mining, and the format of the results that must be transformable to BPMN [11] for allowing the evaluation of the discovered models. Moreover, the results in [3] showed that ILP, HM and Genetic Miner [2] have good performance especially with real-life

¹ <http://www.promtools.org/>.

logs. Thus ETM was selected as evolution of the Genetic miner, while IM, which is presented after the considered study, was selected because it outperforms the other three algorithms according to its authors [7]. Finally the α -algorithm has been considered as the reference for the minimum level of performance.

The **α -algorithm** main aim is to investigate the relationship between transitions for reconstructing causality from a set of sequences of events [17]. Even if the application of the algorithm is quite simple it has a relevant problem to deal with noise, i.e. rare and infrequent behavior (outliers), and incompleteness, i.e. event logs containing only a fragment of all possible behaviors, namely too few events [14]. Moreover, the α -algorithm cannot deal with short loops, i.e. loops of length one or two, non-free-choice constructs and invisible and duplicated tasks.

The **HeuristicsMiner** can be considered an extension of the α -algorithm which takes into account frequencies of events and sequences in the log [20]. To do so, the algorithm only considers the order of the events within a case, while the order of events among cases is not contemplated. The ordering is determined by the timestamp of the activities in the log. The algorithm starts finding the dependency relations between activities and the construction of the dependency graph; then, for each activity, input and output expressions (group of connected activities that precede or follow the concerned activity) are defined, and, finally, long distance dependency are searched. The heuristic approach of the algorithm abstracts from exceptional behaviors and noise (removing edges) making such algorithm more suitable for many real-life logs.

The **ILP Miner** relies on concepts from the language-based theory of regions [8, 18]. Such theory allows to derive a Petri net starting from different classes of languages. However, it seems to be not appropriate when it is directly applied to the field of process discovery. One of the main issues is the size of the resulting Petri net that is strictly (exponentially) dependent on the size of the log. To overcome this issue, the authors combined the ideas from theory of regions with Integer Linear Programming and used the causality relation introduced in the α -algorithm.

The **Inductive Miner** is an algorithm based on a divide-and-conquer approach [6, 7]. Such approach is applied to the log splitting it in sub-logs and then recursively applied to these sub-logs until they contain only a single activity. In this way the problem of discovering a process model for a log is broken down in discovering several sub-processes, one for each sub-log. The algorithm ensures to return a sound, fitting and block-structured process model in finite time. Additionally the authors identified the conditions required to return a model that is language-equivalent to the model of the original process that generated the log.

The **Evolutionary Tree Miner** is a genetic process mining algorithm which allows the user to influence the discovery process based on preferences respect to the four quality dimensions described above [1]. This algorithm uses process trees as model representation which guarantees that all discovered models are sound, i.e. models without deadlocks, livelocks or other anomalies [16]. Like most of the genetic algorithms, ETM randomly creates an initial population then, according to a fitness function, it selects the best individuals on which random mutation

and crossover operations are applied for evolution. Such steps are repeated till the model reaches one of the stop criteria. The algorithm, however, does not ensure the reproducibility of the model: if it is applied multiple times to the same log, the discovered model will most likely be different.

4 Process Mining into Practice

The complexity of the scenario is mainly given by the wide range of coffee machines, which are all assembled on interchangeable production lines. The use of process mining allows an in-depth analysis and representation of the production. It is possible to find out the deviation compared with the expected process model. In this work we considered 450.000 event logs related to six years of production of 32 different coffee machines types. Logs were converted from relational format to eXtensible Event Stream (XES) [4], an XML-based data format for processing event logs and natively supported by ProM. For each type of coffee machine was created a .xes file that is used as input for the five discovery algorithms: α -algorithm, Heuristics Miner, Integer Linear Programming Miner, Inductive Miner, Evolutionary Tree Miner.

This section provides criteria to evaluate the quality of process models discovered by mining algorithms.

4.1 Evaluation Criteria

In order to generate a process model in line with reality, mining algorithms should maintain a proper balance between overfitting and underfitting [14]. Overfitting means that the generated model is too specific and it only admits a behavior similar to the one observed. Underfitting means that the generated model is too general and it also accepts behaviors unrelated to the observed one. A “good” discovered process model needs a proper balance between underfitting and overfitting properties. To asses them we refer to four quality dimensions: *replay fitness*, *precision*, *generalization* and *simplicity* [14]. More in detail, *replay fitness* expresses the portion of the log behavior that can be replayed by the process model, *precision* is the measure of the level of underfitting, i.e. a poor precision means that a model admits unusual behaviors than those shown in the logs, conversely *generalization* is the measure of overfitting, i.e. an high generalization allows also behaviors not seen in the log (maybe not yet observed), while *simplicity* evaluates how easily a human interprets the process model. *Simplicity* could be subject to different interpretation, therefore we considered some complexity metrics [9] to perform an objective assessment. They are following reported.

- *Size* represents the number of nodes (activities and connectors) of the model.
- *Density* (Δ) represents the ratio between the total number of arcs and the maximum possible number of arcs for the same number of nodes.
- *Coefficient of Network Connectivity (CNC)* represents the ratio between arcs and nodes.

- *Average Connector Degree (ACD)* represents the number of nodes a connector is in average connected to.
- *Connector Mismatch (MM)* represents the sum of mismatches for each connector type, where a mismatch is the difference between the total number of outgoing edges from split connectors and the total number of incoming edges at join connectors.
- *Control-Flow Complexity (CFC)* represents the sum over all connectors weighted by their potential combinations of states after a split.

Generally, process models with higher values for such metrics are less understandable and more error-prone as empirical studies have shown [10,12,13].

4.2 Preliminary Results

For reasons of simplicity the logs are grouped for different types of coffee machines and they are enumerated from 1 to 32. For each log set we run the mining using the five algorithms and then we compare quality criteria on the resulting models. Figure 1 shows 5 charts, one for each process mining algorithm, in which values of *fitness*, *precision* and *generalization*, and the corresponding averages are shown. We can observe that independently from the used algorithm, the log sets related to coffee machine types from 1 to 5 and from 20 to 25 show high values for *fitness* and *precision*, but no *generalization*. Some log sets, such as 16 and 29 for example, show in particular the inefficiency of α -algorithm and ILP, while IM, HM and ETM have high performance. The log set related to coffee machine 7 shows a different behavior: not only α -algorithm and ILP have poor results, but also HM underperforms returning values even worse than α -algorithm.

A more detailed analysis of the results is achieved by taking into account the complexity measures. Table 1 shows all the quality values considering log sets with high variability. In Table 1 the first column contains the name of the log set, the second shows the used mining algorithm and the following columns present calculated measures of quality and complexity as indicated in the table header. Bold text highlights best values for each quality dimension. The evaluation of the best discovered model is not simple. Considering the log set 7 we conclude that ETM presents the best score compared to the other algorithms, it has high values of *fitness*, *precision* and *generalization* and low complexity measures. The choice for mining log set 19 is a bit more difficult. In this case HM has the highest *generalization*, while *fitness* and *precision* are very close to best values, but the complexity level is higher than the model resulting from the use of other algorithms. Considering that ILP, IM and ETM have the same level of *fitness*, *precision* and *generalization*, if we penalize models with bigger size, such as in our case, the best algorithm is between ILP and IM. They differ in the ACD and MM measures. Avoiding inconsistencies in the model we prefer to consider readable model, therefore, IM is the most suitable algorithm for log set 19.

More general, we determined the maximum, minimum, average and standard deviation of *fitness*, *precision* and *generalization* in order to obtain the

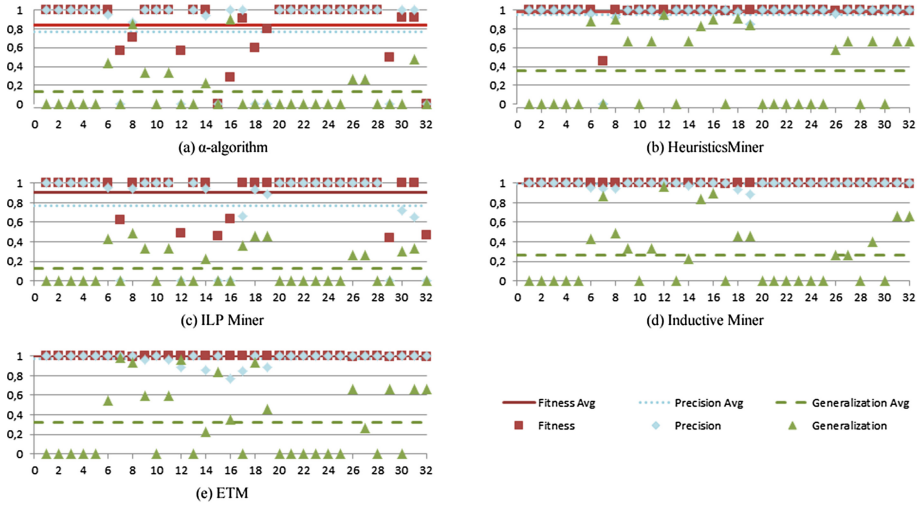


Fig. 1. Process mining results

overall trend of the five mining algorithms. Such values are reported in Table 2. All algorithms show very high maximum values. The only one exception is the *generalization* for ILP. It means that the algorithms have good performance for some logs. Considering the minimum value, good performance for *fitness* and *precision* are confirmed only by IM and ETM. These two algorithms also have high average values and SD very low for *fitness* and *precision* quality dimensions, which shows singular values very close. HM also has a very high average, but the SD indicates a higher variability compared to IM and ETM. Regarding *generalization*, the average level is rather low.

In conclusion, considering quality dimensions we observe that log sets without or with low noise show, in many cases, higher values, albeit slightly, of *fitness* and *precision* in model discovered with α -algorithm than those mined with ETM or HM, whereas the *generalization* is clearly in favor of the latter techniques.

4.3 Discussion and Comparison

A first remark is that no algorithm provides a single log set with the highest values for all the three quality dimensions. Another general remark is about the execution time. ETM is the only algorithm that requires several minutes to discover a model, all the others are significantly faster and they complete in a few seconds.

Complexity measures are not very relevant when considered individually [9]. In order to obtain a better assessment, we estimated all values at once. Furthermore, since the maximum number of activities in our models are 6, complexity measures rarely have very high values, therefore even small differences between two models may be considered relevant. In addition, we point out that the smaller

Table 1. Quality and complexity measures of some discovered models

		Fitness	Precision	Generalization	Size	Δ	CNC	ACD	MM	CFC
7	α	0.564187531	0	0	10	0.18889	1.7	4.5	7	4
	HM	0.454466469	0	0	9	0.13889	1.11111	3	2	2
	ILP	0.627994965	0	0	11	0.15455	1.54545	4.333	6	4
	IM	0.999820965	0.94658	0.86624	10	0.12222	1.1	4	0	1
	ETM	0.998522961	1	0.98718	8	0.125	0.875	0	0	0
8	α	0.714285714	0.875	0.85417	11	0.10909	1.09091	3	2	2
	HM	0.998500664	0.92188	0.90093	10	0.12222	1.1	3	4	4
	ILP	1	0.94444	0.49259	11	0.10909	1.09091	3	2	1
	IM	1	0.94444	0.49259	11	0.10909	1.09091	3	2	1
	ETM	0.997286013	1	0.93333	8	0.125	0.875	0	0	0
12	α	0.564394626	0	0	9	0.22222	1.77778	5	4	1
	HM	0.998831267	0.94444	0.95586	9	0.125	1	3	2	2
	ILP	0.485082741	0	0	13	0.11538	1.38462	3.4	2	5
	IM	0.99833442	1	0.96429	8	0.125	0.875	0	0	0
	ETM	0.998418906	0.88462	0.96526	9	0.125	1	3	2	2
16	α	0.285714286	1	0.9	14	0.09341	1.21429	3.167	13	5
	HM	0.999471858	1	0.9	8	0.125	0.875	0	0	0
	ILP	0.632302405	0	0	11	0.12727	1.27273	3.333	1	1
	IM	0.999471858	1	0.9	8	0.125	0.875	0	0	0
	ETM	1	0.77222	0.35	10	0.12222	1.1	4	0	1
19	α	0.796564669	0	0	10	0.13333	1.2	3	2	3
	HM	0.998594882	0.84848	0.83983	9	0.13889	1.1111	3	4	4
	ILP	1	0.88333	0.45778	9	0.13889	1.1111	3.5	1	1
	IM	1	0.88333	0.45778	9	0.13889	1.1111	4	0	1
	ETM	1	0.88333	0.45778	10	0.12222	1.1	3.333	1	1
29	α	0.491776479	0	0	8	0.26786	1.875	6	5	1
	HM	0.99682937	1	0.66667	7	0.14286	0.85714	0	0	0
	ILP	0.438372793	0	0	8	0.16071	1.125	3	2	0
	IM	0.999582812	1	0.4	8	0.14286	1	3	2	0
	ETM	0.99682937	1	0.66667	7	0.14286	0.85714	0	0	0

models usually have higher density [10]. According to these guidelines, the simplest models are those mined with ETM, HM and IM, α -algorithm, once again has the worst performance.

A final consideration concerns the possible relationship between complexity and quality measures. The evaluation does not permit any conclusion since there are models having a similar level of complexity and very different quality values, e.g. ILP and IM on log set 29 in Table 1, and other models with similar quality values and different complexity.

In summary, after an overall evaluation of all metrics, we can say that the most suitable mining algorithms in our context are IM and ETM. They result with the highest values of fitness and precision for each log set. Considering the average values IM is a bit better than ETM, and similar values can be observed for the generalization, in such case ETM is slightly better, then a new trace may

be more readily accepted by models discovered by ETM than by IM. Also HM presents high values of fitness and precision for most of the log, but on those with high noise, e.g. Log 7, it underperforms. Since the noise in the event logs may increase as time goes, we preferred not to take into account an algorithm with such behavior. ILP and α -algorithm are least suitable because both underperform in all quality dimensions. With regard to the degree of simplicity, the algorithms are almost equivalent for the models corresponding to log without or with low noise, therefore the comparison was made on the remaining models. ILP and α -algorithm show significantly high complexity values, in some cases nearly twice those of other algorithms, so they remain consistent with poor performance. IM and ETM are comparable. They have similar values for most of the log set and present slightly different measures for those that are not similar. Models discovered by HM are on average a bit simpler than those of IM and ETM. We conclude that we preferred IM over ETM due to its performance.

Table 2. Aggregate values for quality dimensions.

	Min.	Max.	Avg.	SD	
Fitness	α	0	1	0.836593796	0.284922192
	HM	0.45447	1	0.981982191	0.094753898
	ILP	0.438377	1	0.909793346	0.190947749
	IM	0.99375	1	0.999559383	0.001182631
	ETM	0.99493	1	0.999250868	0.001191822
Precision	α	0	1	0.774045	0.410313551
	HM	0	1	0.956853125	0.174626942
	ILP	0	1	0.7719325	0.381931605
	IM	0.88333	1	0.988434375	0.026802914
	ETM	0.77222	1	0.974649375	0.056454291
Generalization	α	0	0.9	0.127684688	0.240198833
	HM	0	0.95586	0.35814125	0.390181461
	ILP	0	0.49259	0.133032188	0.179213299
	IM	0	0.96429	0.267528125	0.316556286
	ETM	0	0.98718	0.32354	0.36362457

5 Conclusion and Future Work

In a competitive globalized market, manufacturing companies are forced to continuous improvements to advance. A relevant aspect is that manufacturing is characterized by quite complex production processes. In order to adjust such processes to new requirements a better understanding of the actual processes is needed. The concept of process mining provided appropriate techniques to achieve that purpose.

In this paper, we presented a benchmarking of five process mining algorithms to choose the most appropriate to the motivating case study. α -algorithm, HM,

ILP, IM ed ETM were applied. We then compared such mining algorithms using *fitness*, *precision* and *generalization* values and complexity metrics, for *simplicity*, of the discovered models. Models mined by HM, IM and ETM had high fitness, high precision and low complexity. At the end, the algorithm with the best comprehensive assessment is IM and therefore it is the most suitable for our purposes. Similar studies have already been performed while not in manufacturing. In [3] is presented a comprehensive experimental analysis on the quality of a broad range of mining algorithms: the study reveals that the HM provides the best outcomes, and this is consistent with the findings in our work if we exclude the IM and ETM which have not been taken into account in such study. The outcomes of our work are also validated by [7], where the comparison of mining algorithms, except α , shows that IM and ETM have the best results.

The results of this work have been used by the company for further analysis of the production process. This activity (analysis) revealed that only a small portion of non-standard traces depends on run-time errors of ASCCO. The remaining non-standard traces are due to incorrect managing on the production line of the procedures for fixing or replacing faulty components identified from testing, and of the implementation of some special, and infrequent, customizations. This awareness has driven the company to plan the reorganization of part of the production processes, in order to include such behaviors, and the consequent update of ASCCO for managing these changes.

This paper is part of a larger project whose aim is the use of process mining techniques to support the Business Process Evolution in production chain. The idea is to extend the current traceability system, ASCCO, in order to automatically extract logs, use Inductive Miner to discover the process model, and manage the evolution of such processes through version management approach.

Acknowledgments. We thank Nuova Simonelli and e-Lios for the fruitful collaboration in the project. In particular, we are indebted to Nuova Simonelli President Nando Ottavi for its support and Mauro Parrini who helped us in preparing and conducting this research.

References

1. Buijs, J., van Dongen, B., van der Aalst, W.: A genetic algorithm for discovering process trees. In: 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE, June 2012
2. de Medeiros, A., Weijters, A., van der Aalst, W.M.: Genetic process mining: an experimental evaluation. *Data Min. Knowl. Discov.* **14**(2), 245–304 (2007)
3. De Weerd, J., De Backer, M., Vanthienen, J., Baesens, B.: A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs. *Inf. Syst.* **37**(7), 654–676 (2012)
4. Günther, C.W., Verbeek, E.: XES Standard Definition version 2.0 (2014)
5. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) *Business Process Management Workshops. LNBIP*, vol. 99, pp. 169–194. Springer, Heidelberg (2012)

6. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from event logs - a constructive approach. In: Colom, J.-M., Desel, J. (eds.) *PETRI NETS 2013*. LNCS, vol. 7927, pp. 311–329. Springer, Heidelberg (2013)
7. Leemans, S.J., Fahland, D., van der Aalst, W.M.: Discovering block-structured process models from event logs containing infrequent behaviour. In: Lohmann, N., Song, M., Wohed, P. (eds.) *Business Process Management Workshops*. LNBIP, vol. 171, pp. 66–78. Springer, Heidelberg (2014)
8. Lorenz, R., Mauser, S., Juhás, G., How to synthesize nets from languages: a survey. In: *Proceedings of the Winter Simulation Conference (WSC) 2007*, WSC 2007, pp. 637–647, Piscataway, NJ, USA. IEEE Press (2007)
9. Mendling, J.: *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*. LNBIP, vol. 6. Springer, Heidelberg (2008)
10. Mendling, J., Reijers, H.A., Cardoso, J.: What makes process models understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) *BPM 2007*. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007)
11. OMG. *Business Process Modeling Notation (BPMN)*
12. Reijers, H., Mendling, J.: A study into the factors that influence the understandability of business process models. *IEEE Trans. Syst. Man Cybern. Part A* **41**(3), 449–462 (2011)
13. Rolón, E., Cardoso, J., García, F., Ruiz, F., Piattini, M.: Analysis and validation of control-flow complexity measures with BPMN process models. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) *Enterprise, Business-Process and Information Systems Modeling*. LNBIP, vol. 29, pp. 58–70. Springer, Heidelberg (2009)
14. van der Aalst, W.M.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer, Heidelberg (2011)
15. van der Aalst, W.M.: Processes, no knowledge without : process mining as a tool to find out what people and organizations really do. In: Fred, A., Filipe, J., Dietz, J., Aveiro, D., Liu, K. (eds.) *Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2014)*, pp. 11–16, Rome (2014)
16. van der Aalst, W.M., van Hee, K.M., ter Hofstede, A.H., Sidorova, N., Verbeek, H., Voorhoeve, M., Wynn, M.: Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects Comput.* **23**(3), 333–363 (2011)
17. van der Aalst, W.M., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9), 1128–1142 (2004)
18. van der Werf, J.M.E., van Dongen, B.F., Hurkens, C.A., Serebrenik, A.: Process discovery using integer linear programming. *Fundamenta Informaticae* **94**(3–4), 387–412 (2009)
19. van Dongen, B.F., de Medeiros, A., Verbeek, H.M.W., Weijters, A., van der Aalst, W.M.: The ProM framework: a new era in process mining tool support. In: Ciardo, G., Darondeau, P. (eds.) *Application and Theory of Petri Nets 2005*. LNCS, vol. 3536, pp. 444–454. Springer, Heidelberg (2005)
20. Weijters, A., van der Aalst, W.M., de Medeiros, A.: *Process Mining with the Heuristics Miner Algorithm* (2006)