A User Evaluation of Process Discovery Algorithms in a Software Engineering Company

Simone Agostinelli
Sapienza Universitá di Roma
Rome, Italy
agostinelli@diag.uniroma1.it

Fabrizio Maria Maggi
University of Tartu
Tartu, Estonia
f.m.maggi@ut.ee

Andrea Marrella
Sapienza Universitá di Roma
Rome, Italy
marrella@diag.uniroma1.it

Fredrik Milani
University of Tartu
Tartu, Estonia
milani@ut.ee

***Abstract*—Process mining methods allow analysts to use logs of historical executions of business processes in order to gain knowledge about the actual behavior of these processes. One of the most widely studied process mining operations is automated process discovery. An event log is taken as input by an automated process discovery method and produces a business process model as output that captures the control-flow relations between tasks that are described by the event log. In this setting, this paper provides a systematic comparative evaluation of existing implementations of automated process discovery methods with domain experts by using a real-life event log extracted from an international software engineering company and four quality metrics: understandability, correctness, precision, and usefulness. The evaluation results highlight gaps and unexplored trade-offs in the field and allow researchers to improve the lacks in the automated process discovery methods in terms of usability of process discovery techniques in industry.**

I. Introduction

Today's competitive business environment combined with digital technologies pose a choice to companies, namely to improve or fade away. To improve, they need to increase the efficiencies of their business processes. For decades, analysts have relied on manually modeling the processes and analyzing them so to identify improvement opportunities. Nowadays, by combining business process thinking and data analytics into *process mining* techniques, companies are in a position to take process analysis and improvement to new levels.

According to [1], process mining can be divided into three main branches, process discovery [2]–[4], conformance checking [5]–[9] and process enhancement [10]–[13]. Process discovery has been and remains the most common and widely studied use case [1]. With an event log (capturing unique case ids, activities, and timestamps) as input, every process discovery method produces a business process model. Over the past decade, impressive advancements have been made in this field [1]. Despite this, automated process discovery methods suffer from two recurrent deficiencies when applied to real-life logs [14]: (*i*) they produce large and spaghetti-like models; and (*ii*) they produce models that do not manage to find the right trade-off of the four quality metrics [15]: fitness, generalization, precision, and simplicity. If such models are difficult to understand or perceived as imprecise, they fail to become the valuable tool they are designed to be.

The evaluation of discovery algorithms is generally done by using logs (most commonly real-life industry logs) where the generated models are assessed using different metrics. Nonetheless, the models are rarely evaluated by the process participants or domain experts. In the majority of process mining works [1], the quality of models is often measured from a very theoretical point of view completely ignoring the end user of the techniques developed. In this paper, we aim at focusing on a different form of model evaluation.

In light of this, we seek to investigate how domain experts view and perceive process models *produced* by process discovery algorithms. To the best of our knowledge, the only work similar to ours is the recent preprint published by Bru and Claes [16], which evaluates the quality of process discovery tools (in terms of intuitiveness of the user interface, amount of available features, price, etc.) on the basis of the perceptions of the end users. However, in [16], an analysis of the perceived quality of the models produced by such tools is neglected.

The research question of this paper is therefore "*which discovery algorithm is perceived as the best one by domain experts?*". This question is answered referring to the specific case of a software engineering company developing ERP software. In so doing, relying on a systematic literature review we published in 2019 [4], we have first identified the main discovery algorithms available today. Then, we applied a selection of such algorithms on an event log recorded by the issue tracking system of the company under examination. The models produced were then evaluated using surveys and interviews with the domain experts of that company.

The rest of the paper is structured as follows. Section II briefly presents the structure and results of the systematic literature review, whose findings have been used to select the approaches used for the evaluation. Section III discusses set up and methodology of the evaluation, while Sections IV and V discuss evaluation results and threats to validity. Finally, Section VI concludes the paper and spells out directions for future work.

II. Systematic Literature Review

In order to identify relevant studies and methods related to process discovery, we conducted a *Systematic Literature Review* (SLR) through a rigorous and replicable approach as specified by Kitchenham [17]. In this paper, we briefly present the major steps of our SLR and the most interesting results derived from its enactment. For a detailed discussion, interested readers can refer to [4].

(1) Formulation of the research questions. We scoped the search by formulating five research questions aimed at: *(i)* identifying existing studies proposing methods to perform process discovery; *(ii)* categorizing the output of a method based on the type of process model discovered (i.e., imperative, declarative or hybrid), and the specific language employed (e.g., Petri nets, BPMN, Declare); *(iii)* delving into the specific language constructs supported by a method (e.g., exclusive choice, parallelism, loops); *(iv)* exploring what tool support the different methods have; and *(v)* investigating how the methods have been evaluated.

(2) Search strings definition and data sources selection. Next, we defined four search strings by building combinations of the following keywords: *(i)* "process discovery"; *(ii)* "workflow discovery"; *(iii)* "process learning"; and *(iv)* "workflow learning". We applied each search string to seven popular academic databases: Scopus, Web of Science, IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect and Google Scholar, and retrieved studies based on the occurrence of one of the search strings in the title, the keywords or the abstract of a paper. The search was completed in December 2017.

(3) Definition of inclusion criteria. To ensure an unbiased selection of relevant studies, we defined inclusion criteria, which allowed us to retain studies that: *(i)* propose a method for process discovery from event logs; *(ii)* propose a method that has been implemented and evaluated; *(iii)* are peer-reviewed, written in English and published in 2011 or later (earlier studies were evaluated by De Weerdt in [14]).

(4) Study selection. We analyzed title, abstract, introduction, conclusion and evaluation of the potential relevant studies obtained by the search strings in order to exclude those ones that were clearly not compliant with the inclusion criteria. As a result, we found 86 studies matching all the inclusion criteria. However, many of these studies refer to the same process discovery method, i.e., some studies are either extensions or optimizations of other studies. For such reason, we decided to group the studies by either the last version or the most general one. At the end of this process, 35 main groups of discovery methods were identified.

(5) Study classification. Driven by the research questions, we also classified the methods underlying these studies on the basis of the following dimensions: *(i)* model type (procedural, declarative, hybrid) and language supported (e.g., Petri nets, BPMN, Declare); *(ii)* semantics captured in procedural models (e.g., parallelism, exclusive/inclusive choice and loops); *(iii)* type of implementation (standalone or plug-in) and tool accessibility; *(iv)* type of evaluation data (real-life, synthetic or artificial logs); and *(v)* domain of application (e.g., insurance, banking, healthcare, etc.). Collectively, this information is summarized in Table 2 of [4], where each entry refers to the main study of the 35 groups found.

III. Methodology and Evaluation

In this section, we describe set up and method of our evaluation. In Section III-A, we give a general description of the log and specify the list of miners used. In Section III-B, we describe the preprocessing applied to the original log to create a refined dataset to be used for model discovery. In Section III-C, we specify the setup for the user evaluation. In Section III-D, we present the instruments for conducting our statistical analysis.

A. Experimental setting

In our evaluation, we used a log recorded by the issue tracking system of a company developing ERP software. The log contains data spanning over one year. It has $52\,629$ events and 5551 cases. The log contains information that is used for functional enhancement and bug fixing. The original dataset does not contain explicit information about the activities performed. Therefore, in Section III-B, we describe how this information was extracted from the original log to make it suitable for process discovery. After applying this procedure, 29 unique activities were identified.

In the evaluation, we used a selection of the methods surveyed in [4] (see also Section II). Assessing all the methods that resulted from the literature review would not be possible due to the heterogeneous nature of the outputs produced. Hence, we decided to focus on the largest subset of comparable methods. The methods considered were the ones satisfying the following criteria:

- an implementation of the method is publicly accessible;
- the output of the method is a BPMN model or a model seamlessly convertible into BPMN (i.e., process trees and Petri nets).

The second criterion is a requirement dictated by the fact that the evaluation was performed with business users from industry, which are often non-expert of the technical base formalisms used in the BPM field. Using these criteria, the following miners were identified:

- *alpha$-algorithm* [18], which can discover invisible tasks involved in non-free-choice constructs. The algorithm is an extension of the well-known α algorithm, one of the very first algorithms for automated process discovery, originally presented in [2].
- *BPMN Miner* [19], which is a method for the automated discovery of BPMN models containing sub-processes, activity markers such as multi-instance and loops, and interrupting and non-interrupting boundary events (to model exception handling). The method is robust to noise in event logs.
- *Causal Net Miner* [20], which encodes causal relations gathered from an event log and, if available, from a

TABLE I
COMPLEXITY OF THE DISCOVERED MODELS BEFORE FILTERING

Miner	Size	CNC	Density
alpha$	145	1.490	0.010
BPMN Miner	25	1.760	0.073
CNM	122	2.155	0.017
ETM	124	1.411	0.011
HILP	65	1.600	0.025
HM	97	1.990	0.021
IM	42	1.571	0.038
Structured Miner	54	1.982	0.037

TABLE II
COMPLEXITY OF THE DISCOVERED MODELS AFTER FILTERING

Miner	Size	CNC	Density
alpha$	71	1.408	0.020
BPMN Miner	15	1.600	0.114
CNM	52	1.411	0.020
ETM	84	**1.274**	**0.015**
HILP	34	1.471	0.045
HM	52	1.865	0.037
IM	**28**	1.429	0.053
Structured Miner	**33**	1.454	0.045

background knowledge in terms of precedence constraints over the topology of the resulting process model. The discovery algorithm is formulated in terms of reasoning problems over precedence constraints.

- *Evolutionary Tree Miner* [21], which is based on a genetic algorithm that allows the user to drive the discovery process based on preferences with respect to the four quality dimensions of the discovered model: fitness, precision, generalization and complexity.
- *Hybrid ILP Miner* [22], [23], which is based on hybrid variable-based regions. Through hybrid variable-based regions, it is possible to vary the number of variables used within the ILP (Integer Linear Programming) problems used to discover the process model. Using a different number of variables has an impact on the average computation time for solving the ILP problem.
- *Heuristics Miner* [24], [25], which is a method that can discover process models containing non-trivial constructs, but with a low degree of block-structuredness. At the same time, the method can cope well with noise in event logs.
- *Inductive Miner* [26], [27], which is based on the extraction of process trees from an event log. It efficiently drops infrequent behavior, still ensuring that the discovered model is behaviorally correct (sound) and highly fitting.
- *Structured Miner* [28], which is an improvement of the Heuristics Miner algorithm that separates the objective of producing accurate models from that of ensuring their structuredness and soundness. Instead of directly discovering a structured process model, the approach first discovers accurate, possibly unstructured (and unsound) process models, and then transforms the resulting process model into a structured (and sound) one.

B. From the event log to the process models

As mentioned in section III-A, the log was preprocessed for the evaluation. The activities were not explicitly recorded in the log, but the information about the actors/process participants was. By using the role of the actors, such as *EE Senior Coder*, the activity could be deduced. This step was conducted together with the company to ensure that the activities were correctly captured.

When applying the identified methods to the original log, all the BPMN models discovered were highly complex and spaghetti-like. In TABLE I, we show the metrics measuring the complexity of the models discovered from the original log. In particular, *size* is the number of nodes, *CNC* is the Coefficient of Network Connectivity (CNC), i.e., the ratio between the number of arcs and the number of nodes and *density* is the ratio between the actual number of arcs and the maximum possible number of arcs in any model with the same number of nodes.

To improve the understandability of the models, we decided to filter the original log to isolate frequent behaviors. In particular, we created nine separate logs ranging from a log containing all behavior, to a log containing the behavior shared by at least 2 cases up to a log containing the behavior shared by at least 9 cases. These logs were then used to produce a BPMN model. In TABLE II, we show the metrics measuring the complexity of the models discovered from the log containing the behavior shared by at least 9 cases. For the user evaluation, we decided to select three models for two reasons.

- First, we wanted to avoid getting meaningless results due relatively low amount of participants. If we would have more than three models, the answers could be distributed in a way where it would not be possible to make any statistically strong conclusions.
- Second, we discarded models that were either flower models or spaghetti-like.

In particular, we selected the BPMN model mined by the Evolutionary Tree Miner represented in Fig. 1 (which is the model with the lowest CNC and density). The other two models chosen were the one obtained by the Structured Miner shown in Fig. 2 and the one generated by the Inductive Miner shown in Fig. 3. These two models are the smallest after the one generated by the BPMN Miner that was discarded since, even if very simple, was very imprecise (a flower model allowing any behavior). For anonymization purposes during the evaluation, we referred to the Evolutionary Tree Miner as model A, the Structured Miner as model B, and the Inductive Miner as model C.

C. Evaluation set-up

The evaluation was conducted in two steps both of which involved the domain experts of the company. In particular, the group of experts included employees from the development teams, product managers, testers, documenters, and all the team leads. In total, 18 domain experts participated in the survey which constitute 72% of the process participants at the company (excluding the administrative staff). In the first

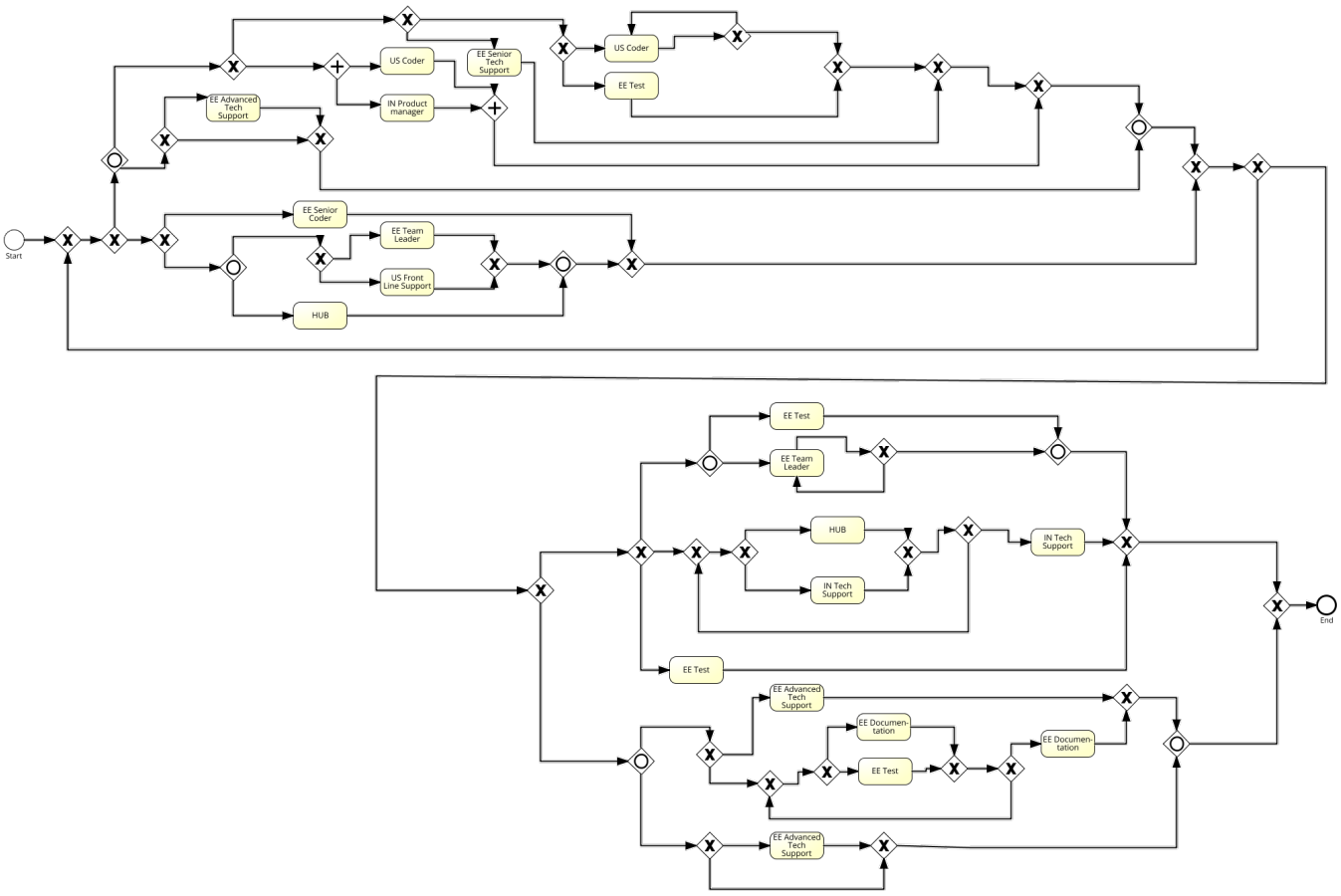

Fig. 1. BPMN model mined by Evolutionary Tree Miner (model A)

step, the participants were given a printed copy of the models and asked to fill out a questionnaire (via Google Forms). The questions asked concerned their familiarity with process models and the time spent working with such models over the past 12 months. In addition, they were asked to compare the models based on different process model quality metrics. Each question was provided in three variants A, B and C representing the specific miner used to create the model. Questions were evaluated using a 7-point Likert scale. The questions asked were the following:

1) Rate how easy it is for you to understand the process model (1 means very difficult, 7 means very easy).
2) Take one path and follow it from the beginning to the end. Rate how easy it is for you to follow your chosen path (1 means very difficult, 7 means very easy).
3) Rate how easy it is for you to distinguish the paths in the model (1 means very difficult, 7 means very easy).
4) Can you recognize any portions of the process you work with in the model? (1 means not at all, 7 means yes, clearly, everything is there).
5) In your estimation, rate how well the model describes your process (1 means that the model is too specific so to exclude some paths that are possible in reality, 7 means that the model is too general so to allow process paths that are not possible in reality).
6) If you were asked to improve your business process, do you find the model useful for this purpose? (1 means useless, 7 means very useful).

The questions correspond to the following process model quality metrics:

- **understandability** - Questions 1, 2 and 3;
- **correctness** - Question 4;
- **precision** - Question 5;
- **usefulness** - Question 6.

In the second step, we carried out a workshop. It was performed in an open form allowing participants to discuss and express their perceptions and to offer qualitative feedback about the models. The discussions did not follow a strict structure, but we used the following targeted questions to moderate the workshop.

- What models are the best ones? Why?
- How did the models look like in general?
- Did the models fulfill your expectations?
- What is missing in the models?
- What could be improved?

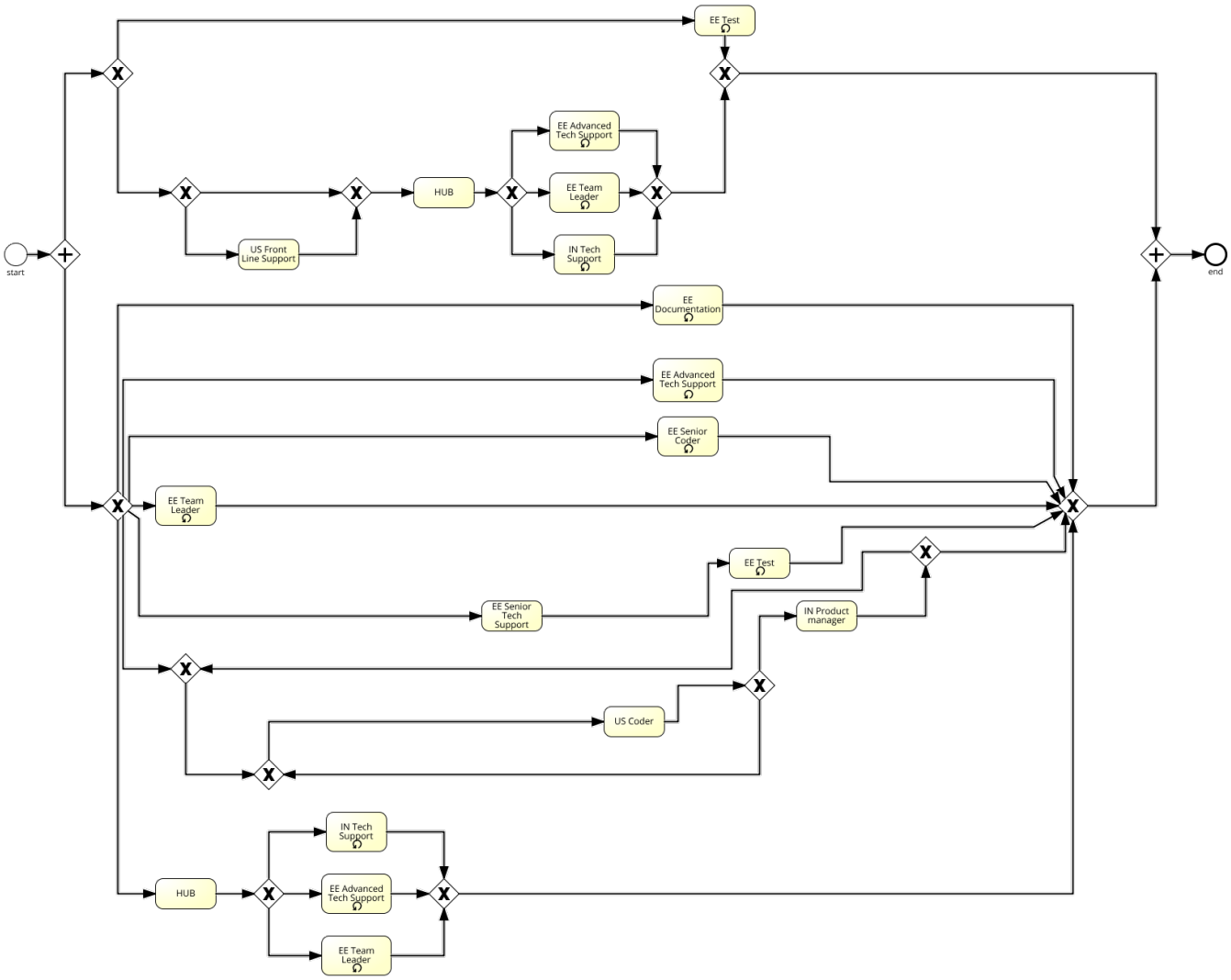

Fig. 2. BPMN model mined by Structured Miner (model B)

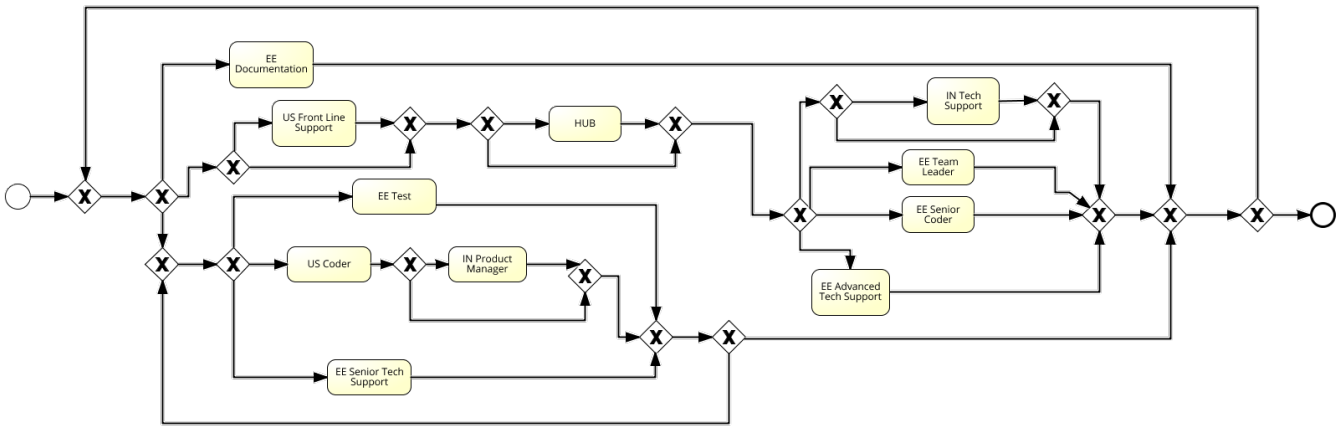

Fig. 3. BPMN model mined by Inductive Miner (model C)

- Would you consider using the discovery algorithms under examination in your company? And automated process discovery in general?

D. Description of statistical analysis instruments

Using statistical analysis, we want to discover if there are any differences in the ratings given by the domain experts to the models. The data analysis was conducted with the free software R.[1] The answers of the questionnaire were extracted from the Google Forms and formatted to match the input format required by R. Since 3 questions correspond to the quality metric *understandability*, for each model, we derived the general rating corresponding to this dimension by averaging over the values obtained for the 3 questions. We conducted the statistical analysis over the 3 groups *model A*, *model B*, and *model C*, each consisting of 4 subgroups, one for each quality metric. We formulated the following hypotheses:

- **The null hypothesis**: There is no difference in the mean ratings of the models.
- **The experimental hypothesis**: There is at least one model that is different from the others.

The hypotheses were tested using two-way ANOVA. Assuming the independence of the observations, to apply the ANOVA test, it was necessary to verify that the residuals are normally distributed and have the same variance (homogeneity of variances) for each combination of the groups. The normality assumption was assessed using the Shapiro-Wilk test and QQ-plots, while the homogeneity of variances was assessed with the Levene's test. Violin plots were used to represent the results. A violin plot is a method of plotting numeric data including a marker for specifying the median of the data and a box indicating the interquartile range. Overlaid on this box plot there is a kernel density estimation, which is a non-parametric way to estimate the probability density function of a random variable.

IV. Evaluation results

The research question we want to answer is how discovery algorithms are perceived by the domain experts from the software engineering company under examination. To answer this question, in this section, we describe the results obtained in the two steps of the survey conducted with the domain experts from the company under examination, i.e., the results coming from the questionnaire (Section IV-A) and the ones gathered during the workshop (Section IV-B).

A. Questionnaire Results

Table III summarizes the mean values of the ratings given by the domain experts to the models for all the dimensions under examination. The violin plots obtained from the first 3 questions about understandability are shown in Fig. 4. Question 1, *Rate how easy it is for you to understand the process model*, has a mean of 3.28 for model A and 4.72 for models B and C. For what concerns question 2, *Take one path and follow it from the beginning to the end. Rate how easy it is for you to follow your chosen path*, model A has the lowest mean value of 3.61, then model B with 5.11 and model C with the highest mean value of 5.39. The mean values for question 3, *Rate how easy it is for you to distinguish the paths in the model*, are 3.22 for model A, 5.39 for model B, and 5.22 for model C. To summarize, model A is perceived as the least understandable for all three questions, whereas models B and C are comparable.

TABLE III
Complexity measures and mean ratings of the models

	Model A	Model B	Model C
Size	84	33	28
CNC	1.28	1.45	1.43
Density	0.015	0.045	0.053
Understandability Q1	3.28	4.72	4.72
Understandability Q2	3.61	5.11	5.39
Understandability Q3	3.22	5.39	5.22
Correctness	4.11	4.89	5.28
Precision	4.44	4.44	5.22
Usefulness	3.44	4.17	5.22

Fig. 5 shows the violin plots in terms of perceived correctness of the discovered models, which was investigated with question 4, *Can you recognize any portions of the process you work with in the model?*. The perceived correctness of the three models is comparable with a slight preference for model C over B and A. In particular, the interquartile range and density are better for model C and this is also reflected in the mean values, which are 4.11, 4.89 and 5.28 for models A, B and C, respectively.

Fig. 6 shows the violin plots for question 5, *In your estimation, rate how well the model describes your process*, concerning precision. As can be seen from the figure, the interquartile ranges for models A and B are very similar (and both have a mean value of 4.44), whereas the results for model C show a different distribution of responses. Considering the distribution of the respondents results for model C (with a mean value of 5.22), it seems that model C is perceived to be more general as compared to model A and B.

The final question aimed at assessing the perceived usefulness. As processes are often discovered for process enhance-

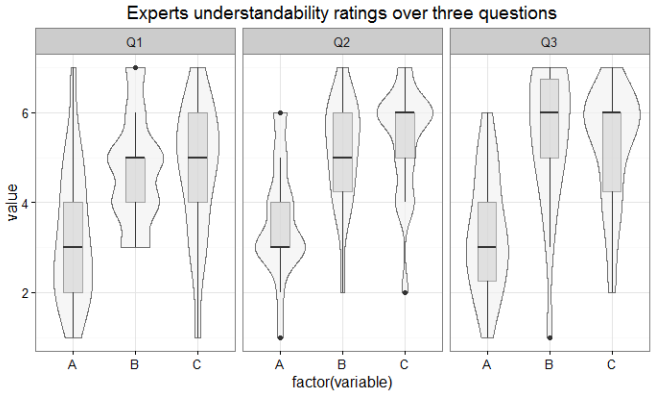

Fig. 4. Results for *understandability* (the bold line denotes the median)

[1]https://www.r-project.org/

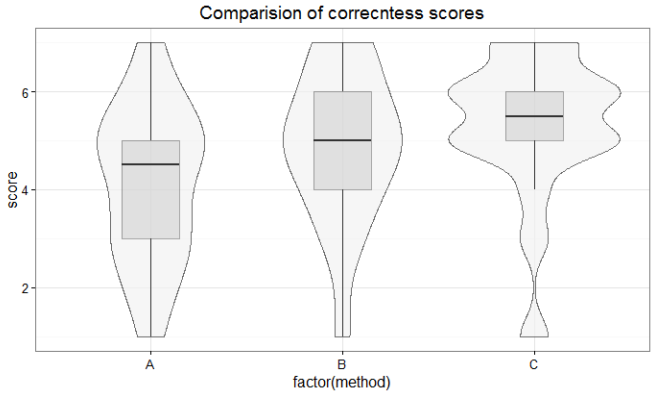

Fig. 5. Results for *correctness* (the bold line denotes the median)

Fig. 6. Results for *precision* (the bold line denotes the median)

ment, the respondents were asked about the usefulness of the models for improving the processes. The results show that

Fig. 7. Results for *usefulness* (the bold line denotes the median)

model C was perceived as more useful as can be seen in Figure 7. The difference between model A as compared with model B and C is significant. Model A has a mean value of 3.44, whereas model B and C have a mean value of 4.17 and 5.22, respectively.

The hypotheses were tested using two-way ANOVA that rejected the null hypothesis (p-value of 9.297e-07) and confirmed that there is at least one model that is different from the others. Therefore, we performed the Tukey HSD test to check the models pairwise and discover statistically significant differences. The result was that there is a statistically significant difference between model A and model B and between model A and model C. However, we could not conclude that there is a statistically significant difference between models B and C.

B. Workshop Results

After the survey, we engaged in discussions with the domain experts about the different models. The discussions were semi-structured and a set of topical questions were asked (listed in Section III-C). It should be noted that the discussions were not about usefulness of process discovery techniques in general. Rather, the aim of the discussions was to better understand how the domain experts perceived the generated models.

The most common observation mentioned by the domain experts concerned the possibility of overlaying the models with additional information. Adding information about path frequency, frequency of activities, and performance metrics would improve the usability and the understandability of the models considerably. The reasons were that such information would allow distinguishing most commonly executed paths, deviations, frequency of deviations and execution times thus adding significant value when using the models for process improvement. The experts also noted that in model A and C, it is possible to reach the end of the process without passing through any activity. As such, since there are no empty traces in the log, it would be relevant to compare the discovered models with the log to see which paths shown in the model are actually present in the log.

It was also noted that the models could be made simpler by providing the user with the possibility of hiding infrequent paths. Furthermore, the domain experts shared that the models contained more gateways than perceived necessary. In this regard, they also saw the need of annotating the gateways with routing probabilities. Using routing probabilities, the number of gateways could be reduced by hiding branches that are taken less frequently. Finally, several experts mentioned that the understandability of the models could be improved by introducing sub-processes. The gradually growing consensus was that the models were fairly accurate, they did capture most of the processes existing in the company, and that model C best reflected the company's everyday work.

C. Summary

Taking all four aspects considered in the comparative evaluation, we note that model B and C are perceived as better as compared to model A. The number of nodes of model A (84) is clearly above the threshold suggested by the literature [29], [30] to ensure understandability. It is therefore not surprising that domain experts found this model to be the least understandable. In regards to correctness, precision and usefulness, the domain experts clearly favor model B and C

over model A. While the values for model C are slightly higher than for B, their difference is not statistically significant.

During the workshop, the domain experts expressed the usefulness of the discovery techniques for understanding the current state of a business process, but wished to see simpler models enhanced with additional information about path frequencies and performance. It should be noted that commercial products for process discovery such as Disco[2] or Celonis[3] and recently also the open source tool Apromore[4] do provide simpler models, overlay the models with frequency and performance information, and allow for filtering based on frequencies of activities and paths.

V. Threats to Validity

The evaluation of our study has some threats to validity. The main one is about external validity, i.e., the extent by which the findings can be generalized beyond the scope of the study [31]. The specific environment in which the evaluation was conducted and the use of one process model do not constitute sufficient base to draw conclusions about the goodness of the discovery algorithms under analysis. Given this limitation, our discussions should be considered as indicative observations rather than conclusive statements.

Secondly, as explained in Section III, we decided to restrict the comparison to three models to ensure that statistically strong conclusions can be made. To achieve this, we discarded models that were either flower models or spaghetti-like. Of course, this does not mean that such discarded models do not reflect the real complex behavior of the recorded process, but just that such kind of models are in general too complex to be evaluated and understood by end users [32].

Another threat to validity is about construct validity. Construct validity considers the extent by which the perception of the domain experts in regards to the quality measures matches what the study wants to evaluate [33]. This threat to validity was mitigated by employing prolonged involvement [31]. Prolonged involvement guarantees a trustful relation between the researchers and the organization. In this case, one of the authors have had collaboration with the company extending over three years prior to conducting this study. The prolonged involvement built trust that motivated participants to spend more time on providing data and feedback. Furthermore, the prolonged involvement also allowed for a better understanding of how participants interpreted the various terms used in the evaluation.

Finally, we employed triangulation [34] to improve the reliability of the results, i.e., we used data from different sources. In our study, we collected evaluation data from both surveys and group interviews.

VI. Conclusion

This paper has presented a comparative evaluation of existing implementations of automated process discovery methods using a real-life event log from an international software engineering company. From our analysis, we discovered that domain experts found the Inductive Miner (model C) and the Structured Miner (model B) to be the best ones. The discussions with the experts suggested that showing path frequencies, routing probabilities, and time performance in the models would represent a significant added value that would help them in finding improvement opportunities.

As future work, we are going to perform a more robust evaluation tackling all the weaknesses listed in Section V. In addition, we aim at extending our analysis to commercial process mining tools (e.g., Disco, Celonis, etc.), which have been discarded in this paper since they do not produce BPMN models in output (they usually discover directly-follows graphs, thus violating the first cut-off criterion discussed in Section III-A). In this regard, the functionality recently introduced in Apromore to translate a directly-follows graph into BPMN and vice versa could support our investigation.

[2]https://fluxicon.com/disco/

[3]https://www.celonis.com

[4]http://apromore.org/

Acknowledgments

The research of Fabrizio Maria Maggi and Fredrik Milani has been partly supported by the Estonian Research Council Grant IUT20-55. The research of Simone Agostinelli and Andrea Marrella has been partly supported by the "Dipartimento di Eccellenza" grant, the H2020 RISE project FIRST (grant #734599), the H2020 ERC project NOTAE (grant #786572), the Sapienza grants IT-SHIRT, ROCKET and METRICS, the Lazio regional initiative "Centro di eccellenza DTC Lazio" and the project ARCA.

References

[1] W. M. P. van der Aalst, *Process Mining: Data science in action*. Springer, 2016.

[2] W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 9, 2004.

[3] B. F. van Dongen, A. K. Alves de Medeiros, and L. Wen, "Process mining: Overview and outlook of Petri net discovery algorithms," in *Transactions on Petri Nets and Other Models of Concurrency II*. Springer, 2009, pp. 225–242.

[4] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, F. M. Maggi, A. Marrella, M. Mecella, and A. Soo, "Automated Discovery of Process Models from Event Logs: Review and Benchmark," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 686–705, 2019. [Online]. Available: https://doi.org/10.1109/TKDE.2018.2841877

[5] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, 2012.

[6] G. De Giacomo, F. M. Maggi, A. Marrella, and F. Patrizi, "On the Disruptive Effectiveness of Automated Planning for LTL$_f$-Based Trace Alignment," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017, pp. 3555–3561.

[7] M. de Leoni and A. Marrella, "Aligning Real Process Executions and Prescriptive Process Models through Automated Planning," *Expert Syst. Appl.*, vol. 82, pp. 162–183, 2017.

[8] G. De Giacomo, F. M. Maggi, A. Marrella, and S. Sardiña, "Computing Trace Alignment against Declarative Process Models through Planning," in *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016*, 2016, pp. 367–375.

[9] M. de Leoni, G. Lanciano, and A. Marrella, "Aligning Partially-Ordered Process-Execution Traces and Models Using Automated Planning," in *Proceedings of the Twenty-Eight International Conference on Automated Planning and Scheduling (ICAPS 2018)*, 2018, pp. 321–329. [Online]. Available: https://aaai.org/ocs/index.php/ICAPS/ICAPS18/paper/view/17739/16951

[10] F. M. Maggi, D. Corapi, A. Russo, E. Lupu, and G. Visaggio, "Revising process models through inductive learning," in *Business Process Management Workshops*, M. zur Muehlen and J. Su, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 182–193.

[11] D. Fahland and W. M. P. van der Aalst, "Repairing process models to reflect reality," in *International Conference on Business Process Management*. Springer, 2012, pp. 229–245.

[12] R. P. J. C. Bose, F. M. Maggi, and W. M. P. van der Aalst, "Enhancing Declare maps based on event correlations," in *Business Process Management - 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*, 2013, pp. 97–112.

[13] F. M. Maggi, A. Marrella, G. Capezzuto, and A. Armas Cervantes, "Explaining non-compliance of business process models through automated planning," in *Service-Oriented Computing*. Cham: Springer International Publishing, 2018, pp. 181–197.

[14] J. De Weerdt, M. De Backer, J. Vanthienen, and B. Baesens, "A multi-dimensional quality assessment of state-of-the-art process discovery algorithms using real-life event logs," *Information Systems*, vol. 37, no. 7, pp. 654–676, 2012.

[15] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery," in *OTM Conferences (1)*, vol. 7565, 2012, pp. 305–322.

[16] J. Claes and F. Bru, "The perceived quality of process discovery tools," 2018.

[17] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.

[18] Q. Guo, L. Wen, J. Wang, Z. Yan, and S. Y. Philip, "Mining invisible tasks in non-free-choice constructs," in *International Conference on Business Process Management*. Springer, 2015, pp. 109–125.

[19] R. Conforti, M. Dumas, L. García-Bañuelos, and M. La Rosa, "BPMN Miner: Automated discovery of BPMN process models with hierarchical structure," *Information Systems*, vol. 56, pp. 284–303, 2016.

[20] G. Greco, A. Guzzo, F. Lupia, and L. Pontieri, "Process discovery under precedence constraints," *ACM Trans. on Know. Discovery from Data (TKDD)*, vol. 9, no. 4, p. 32, 2015.

[21] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, "Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity," *International Journal of Cooperative Information Systems*, vol. 23, no. 01, p. 1440001, 2014.

[22] S. J. van Zelst, B. F. van Dongen, and W. M. P. van der Aalst, "ILP-Based Process Discovery Using Hybrid Regions," in *International Workshop on Algorithms & Theories for the Analysis of Event Data, ATAED 2015*, ser. CEUR Workshop Proceedings, vol. 1371. CEUR-WS.org, 2015, pp. 47–61.

[23] S. J. van Zelst, B. F. van Dongen, W. M. P. van der Aalst, and H. M. W. Verbeek, "Discovering Workflow nets using Integer Linear Programming," *Computing*, vol. 100, no. 5, pp. 529–556, 2018.

[24] A. J. M. M. Weijters and J. T. S. Ribeiro, "Flexible heuristics miner (FHM)," in *2011 IEEE Symp. on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2011.

[25] F. Mannhardt, M. de Leoni, H. A. Reijers, and W. M. P. van der Aalst, "Data-Driven Process Discovery-Revealing Conditional Infrequent Behavior from Event Logs," in *International Conference on Advanced Information Systems Engineering*. Springer, 2017, pp. 545–560.

[26] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach," in *Application and Theory of Petri Nets and Concurrency: 34th International Conference, PETRI NETS 2013, Milan, Italy, June 24-28, 2013. Proceedings*. Springer, 2013, pp. 311–329.

[27] ——, "Discovering Block-Structured Process Models from Incomplete Event Logs," in *Application and Theory of Petri Nets and Concurrency: 35th International Conference, PETRI NETS 2014*. Springer, 2014.

[28] A. Augusto, R. Conforti, M. Dumas, and M. La Rosa, "Automated Discovery of Structured Process Models From Event Logs: The Discover-and-Structure Approach," *Data and Knowledge Engineering (to appear)*, 2017.

[29] J. Mendling, G. Neumann, and W. M. P. van der Aalst, "Understanding the occurrence of errors in process models based on metrics," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. Springer, 2007.

[30] J. Mendling, L. Sánchez-González, F. García, and M. La Rosa, "Thresholds for error probability measures of business process models," *Journal of Systems and Software*, vol. 85, no. 5, pp. 1188–1197, 2012.

[31] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

[32] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer, 2013, vol. 1.

[33] S. Zugal, J. Pinggera, and B. Weber, "Assessing process models with cognitive psychology." in *EMISA*, vol. 190, 2011, pp. 177–182.

[34] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, no. 4, pp. 557–572, 1999.