# A New Process Discovery Algorithm for Exploratory Data Analysis

Jonas Lieben
Supervisors: Benoît Depaire and Mieke Jans

Hasselt University Martelarenlaan 42, 3500 Diepenbeek, Belgium
FWO, Egmontstraat 5, 1000 Brussels, Belgium
jonas.lieben@uhasselt.be

**Abstract.** The domain of process mining created many discovery techniques which can be used to generate a process representation of the data. However, existing techniques come with a flaw for exploratory data analysis (EDA). They tend to produce process models which contain more process behaviour than is observed in the data and do not optimize for understandability. This severely limits their value for EDA, because only patterns which can be observed from the data should be distilled when performing an EDA. We explain why this limitation is important and give a methodology to overcome this. This methodology describes how a discovery algorithm can be developed that is suitable for EDA.

**Keywords:** Process, Exploratory data analysis, Comprehensibility, Precision, Process discovery algorithm

## 1 Research Context

During the past years, companies are increasingly storing and collecting event data. This type of data describes the occurrences of events during the execution of a business process. Originally, its main source are IT systems supporting business operations. Recently, Internet of Things, with all its sensors measuring changes in the environment, has become a new important source of event data.

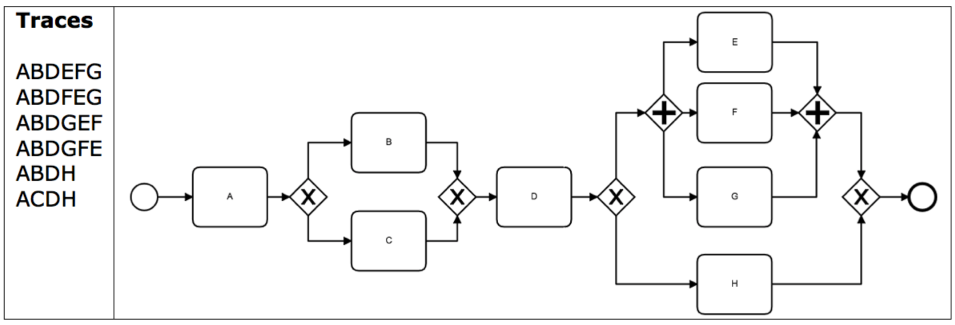
The analysis of event data and the underlying process belongs to the domain of process mining. Within process mining, three broad categories exist: process discovery, conformance checking and process enhancement [1]. This project fits in the subdomain of process discovery. The goal of process discovery is to create a (visual) model representing the process based on event data. These models are typically visualised in a graph based language such as Petri nets or BPMN.

Models learned from data serve multiple purposes. In this project we focus on the purpose to describe and summarise event data and to reveal interesting patterns within this data. Such models are used for exploratory research. Exploratory data analysis (EDA) is an important preliminary to confirmatory and predictive analysis. John Tukey stated that: "Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone as the first step" [17]. When presented with a large event log, EDA provides a good understanding of the data at hand which is essential for a useful further analysis of the underlying process.

Researchers from the domain of process mining proposed many discovery techniques which can be used to create a process representation of event data. However, these techniques come with a limitation for EDA. They tend to generate process models which contain more process behaviour than is observed in the data, because they were developed with the rationale that an event log is incomplete. Therefore, they produce models which generalise the behaviour in the event log to represent all possible behaviour.

Figure 1 shows a simple example process model which was discovered with event data. A process can be executed multiple times and each execution refers to a case. The sequence of the events during the process execution is called a trace. Two cases share the same trace if the events occur in the same order. The left side of Figure 1 shows an example of an event log containing several traces.

The model of figure 1 shows the BPMN representation of the model discovered by Evolutionary Tree Miner [3] using the traces on the left side. Other miners discover similar models. While the model is a concise representation of the event data, it is not perfectly precise. In process mining, a perfectly precise model only contains the observed process behaviour. The model in Figure 1 is not perfectly precise, because it allows the execution of the unobserved traces ACDEFG and ABDEGF. To our knowledge, there are not many process discovery techniques that guarantee a perfectly precise model as outcome.


**Traces**

ABDEFG
ABDFEG
ABDGEF
ABDGFE
ABDH
ACDH

**Fig. 1.** An imprecise model representing a set of traces

We consider this an important research gap for EDA as caution is needed when visualising patterns which are not completely present within the data. Such patterns might mislead the researcher in its conclusions. Based on the model in Figure 1 the researcher might conclude that E, F and G occur in any order. However, the data does not support this conclusion. Close inspection of the data reveals for example that G never occurs between E and F. Therefore, a good exploratory model should only hint at patterns that are not fully supported in the data, but should never present them as facts.

Mining a perfectly precise model is not difficult. The trace model, which consists of a single exclusive choice where each possible path represents a trace from the event log, is always perfectly precise. Figure 2 shows the trace model for the traces in Figure 1.

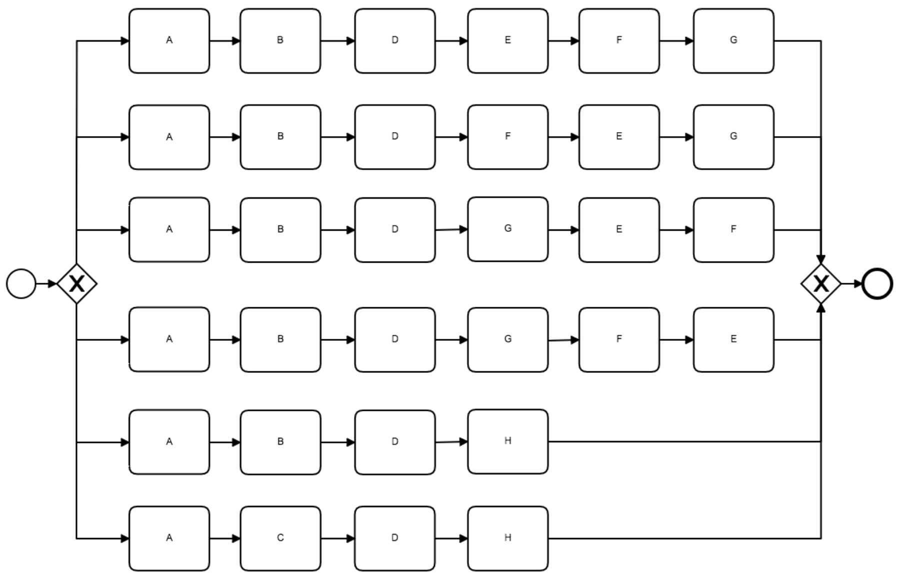


**Fig. 2.** A trace model for the traces

Although the trace model is perfectly precise, it performs poorly in some aspects of comprehensibility, because it is difficult to identify patterns of choice and concurrency. Exploratory models should not only be perfectly precise, but also be optimised for comprehensibility. Figure 3, for example, illustrates a perfectly precise model for the traces in Figure 1, but has a higher comprehensibility than the trace model. The main difference between both models is the number of duplicate tasks. The relation between duplicate tasks and comprehensibility is complex and non-linear. Too many duplicate tasks hide patterns (cfr. Figure 2) and decrease the comprehensibility. However, a certain number of duplicate tasks also adds structure to the process and reduces clutter [13] which increases comprehensibility.

This leads to the second research gap which we will address in this project. According to a recent literature review [5], none of the existing metrics measuring process model comprehension account for the influence of duplicate tasks on comprehensibility. This is due to the fact that it is implicitly assumed that process models have unique labels.

This research project is important because the current discovery algorithms produce imprecise models which limit their value for EDA. As EDA is an impor-

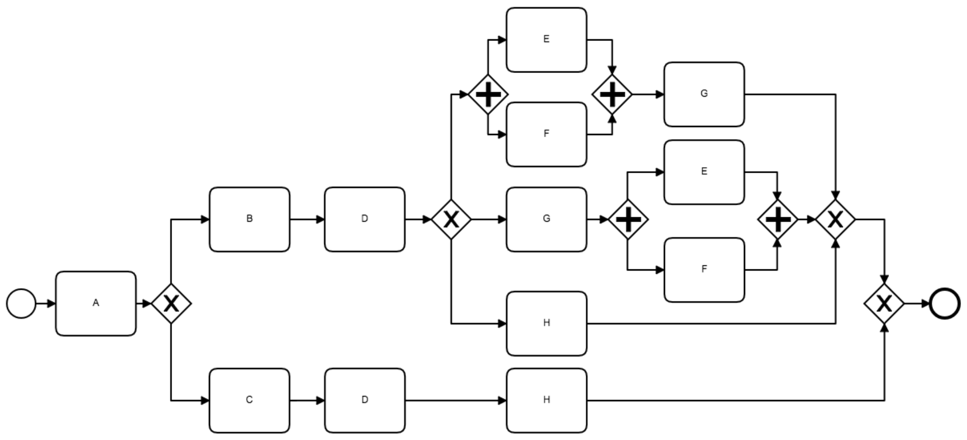


**Fig. 3.** A perfectly precise model with limited duplicate activities

tant first step for any data analysis project, having an algorithm which produces comprehensible and precise models will make it significantly easier to identify interesting patterns and ideas for follow-up (confirmatory/predictive) analysis. Our research is unconventional in the sense that the guiding principles for our discovery algorithm will be precision and comprehensibility, whereas current techniques rely on the assumption that the event log is incomplete.

## 2 Research Objectives

The overall research goal is to develop a process discovery algorithm for exploratory data analysis which generates a perfectly precise and comprehensible process visualization of the event data. To achieve this overall research goal, three research objectives need to be achieved:

Firstly, a discovery algorithm needs to be developed. This algorithm should be able to generate models with perfect precision, optimised comprehensibility and representing a certain number of traces from the event log. In addition to generating perfectly precise models, the algorithm must meet the following requirements to be of value during exploratory research:

- Generate comprehensible models: the purpose of EDA is to get a good understanding of the data and to easily recognise interesting patterns. Optimization of comprehensibility must directly guide the algorithm.
- Generate models representing a certain number of traces: traditional discovery algorithms focus on learning a model for the entire event log, which often results in overly complex models. During EDA, the researcher is not always interested in a single model representing all traces. Therefore, the data analyst should be able to set the number of traces which should be represented by the model. The algorithm should select the set of traces which results in the most comprehensible perfectly precise model.

– Allow for different comprehensibility measures: the algorithm should be flexible enough such that a different measure can be used without changes to the algorithm.
– Be extensible: part of this project will focus on how to optimally visualise certain aspects of a process. These insights will be incorporated into the algorithm. The mechanism to do so must be flexible enough such that future insights can be easily incorporated.
– Use BPMN as the graphical notation: empirical research has shown that the BPMN notation appears to be the strongest in providing for a good understanding by model readers [6].

Secondly, more comprehensible visualizations for partial parallelism and long-term dependencies need to be designed. Partial parallelism occurs when a set of activities seem to happen in parallel, but not all possible combinations are observed in the event log. Long-term dependencies are observed when an exclusive choice is partially determined by the occurrence or non-occurrence of previous activities in the trace. Both constructs are present in the data of Figure 1. Activities E, F and G are only partially in parallel since we never observe a trace with G occuring between E and F. The exclusive choice after activity D is limited to activity H when activity C occurred. Both constructs have a tendency to make a model less comprehensible. The goal of this research objective is to search for different kinds of visualization to improve comprehensibility.

Thirdly, an empirically validated comprehensibility measure needs to be developed. This measure should be applicable to the models generated by our algorithm. Our algorithm uses a comprehensibility measure as its guiding mechanism. This implies that the measure also needs to account for the comprehensibility cost of duplicate tasks and the different visualizations developed as part of research objective two.

## 3 State of the Art

In the domain of process mining, many algorithms have been developed to discover the control-flow of process models. To our knowledge, none of the existing algorithms create perfectly precise models while optimizing for the understandability. Most algorithms put a less stringent notion of completeness than global completeness. The notion of global completeness implies that all possible behaviour of the process is included in the log [4]. As the creators of most existing algorithms made the assumption that the log is incomplete, there are patterns included into the process models which are not present in the log.

Existing discovery algorithms can be categorized into five categories [4]. The first category is the abstraction-based algorithms. One of the best known discovery algorithms is the $\alpha$-algorithm [2]. The $\alpha$-algorithm and its derivatives are all abstraction-based algorithms. The heuristics miner [18] is the only algorithm belonging to the heuristic-based algorithms category and takes into account the presence of noise. The third category is the search-based algorithms, which contains the Evolutionary Tree Miner based on genetic algorithms [3]. This category contains all algorithms which use metaheuristics to infer a process model. Models created by existing algorithms of the first three categories are not perfectly precise, because one of the underlying assumptions is the incompleteness of the event log. Language-based region algorithms can generate perfectly precise models. This fourth category uses the theory of regions to construct a process model [14]. However, the algorithms do not directly optimize for understandability. The last category contains the state discovery algorithms [4]. These algorithms first construct a transition system and then derive a Petri net. Nevertheless, they do not directly optimise for understandability either.

## 4 Research Methodology

The general methodology for this project follows the principles of design science research (DSR). DSR deals with the creation of artifacts and scientific knowledge about these artifacts with the goal to provide solutions to a class of problems [8]. A typical DSR project consists of five steps: problem identification, requirement specification, artifact design and development, artifact evaluation and result communication. The problem identification step has largely been done during the preliminary study in preparation for this research paper. For each research objective of Section 2, a methodology will be described.

### 4.1 The Creation of the Comprehensibility Measure

We start with the development of the empirically validated comprehensibility measure, because the new discovery algorithm can only be created using the measure. The first step in the development of this measure is a literature review to identify different aspects of a process model which have been empirically proven to influence comprehensibility. Through this literature review, we are able to gather the requirements for the measure. Therefore, this step is the requirement specification.

During the artifact design and development phase, we will develop an algorithm which quantifies the presence of these aspects within the process model. This part of the study has already been executed. 23 existing metrics were identified and implemented using the programming language R. The results are published in the form of an R package on CRAN [1] and will be sent to the journal paper SoftwareX. At the moment, there are no other software packages that can calculate all implemented metrics for a batch of BPMN models. In addition, an exploratory factor analysis is performed. This factor analysis allows to discover the underlying dimensions of the large number of metrics. The sample of models used for the factor analysis consisted of BPMN models from the BPM AI (Business Process Management Academic Initiative) [11] and models generated by the PTandLogGenerator [9]. The results of this factor analysis are published

[1] `https://cran.r-project.org/web/packages/understandBPMN/index.html`

as conference proceedings and are presented at the EOMAS (Enterprise & Organizational Modeling and Simulation) workshop which takes place in conjunction with CAISE 2018.

Next, we will conduct an experiment to determine the impact of each dimension on comprehensibility. Participants will receive process models and a set of questions to test their understanding of the models. We apply a within-subjects design to control for the effects on comprehensibility related to the model reader. The dependent variables will be an objective comprehension accuracy measure such as percentage of correct answers, a time-taken measure and a subjective comprehension difficulty measure. The independent variables in this study will be the quantifications of the different model aspects within the models. After running the experiment, we will apply a multi-level regression analysis on the collected data to determine the impact of each factor on comprehensibility. The parameter estimates will become the empirically-validated weights for our new comprehension measure. The artifact will afterwards be evaluated and demonstrated and the results will be communicated as a journal paper.

### 4.2 The Development of the Discovery Algorithm

When the comprehensibility measure is created, the discovery algorithm can be created. Two versions of the discovery algorithm will be developed: one which generates a model representing all traces and one which generates a model representing a predefined minimum number of traces. To develop and design the algorithms, we will transform the discovery problem into an optimization problem. This approach has been applied before by [3], which used genetic algorithms as search strategy. However, genetic algorithms are less suited for our problem since it would be difficult to define mutate and cross-over operators that are necessary to result in perfectly precise models.

Our approach is inspired by Iterated Local Search [15], which has not been applied before in this context. Our algorithm will use the trace model as initial solution and apply domain-specific local search operators (LSOs) to modify the model by transforming duplicate tasks into more complex process structures. For the first version, we will develop at least two LSOs: one for parallel constructs and one for exclusive choice constructs. Other LSOs may be defined later. Each LSO should guarantee perfect precision after transformation. The LSOs are the mechanism that make the algorithm extensible.

For the second version of the algorithm, two new operators will be created: a trace removal and a trace imputation operator. The removal operator will remove parts of the process model that correspond to entire traces. The imputation operator will add entire traces from the event log to the model. Both operators should modify the model in such a way that perfect precision remains guaranteed.

To verify whether our models are perfectly precise and represent all traces, we first use the Behavior Recall metric [7] to check whether all traces are represented by the model. If so, we will use the ETC Precision [12] to test if the model is perfectly precise. Because both metrics require the process model to be represented as a Petri net, we will use the transformation algorithm in [10]. For the BPMN constructs in our models, this algorithm guarantees bisimilarity. To evaluate the comprehensibility of the models, there is no other algorithm to compare with. Therefore, we are limited to a descriptive analysis of the algorithms performance in terms of comprehensibility. We will analyze the improvements with respect to the trace model and apply a sensitivity analysis to see which aspects influence the algorithms ability to improve comprehensibility. For evaluation we will use a broad set of event logs, both real and artificial. The real data will be taken from the collection made available by the IEEE Task Force on Process Mining. The artificial data sets will be created using [9]. The results will be made available in an R package on CRAN and scientific papers.

### 4.3 The Design of Alternative Visualizations for Partial Parallelism and Longterm Dependencies

We aim to create more comprehensible visualizations for partial parallelism and longterm dependencies. To determine the requirements of these alternative visualizations, we will apply a multi-dimensional long-term case study with expert users as suggested in [16]. Since the purpose of the case study is to increase transferability to people who have the same needs, a sample of 3 to 5 expert users is appropriate [16]. Experts will be data analysts, both from academia and industry. The case study is multi-dimensional because it combines different research methods such as interviews and observations. It is also long-term, because it involves a longitudinal study throughout the entire DSR cycle.

These new visualisations need to be incorporated in the comprehensibility measure of Section 4.1 and in the discovery algorithm of Section 4.2.

## 5 Conclusion

Industry is becoming increasingly data-driven. The past decade both the amount of data collected and the nature of the data has changed. This project focusses on event data, which describes how (business) processes are executed. The first step for retrieving insights from data is through exploratory data analysis (EDA). Despite the many algorithms which discover process models from event data, none of them are really suited for EDA for two reasons. Firstly, they tend to create models which contain behaviour that was not observed data. Secondly, almost none of the existing algorithms optimise their models in terms of comprehensibility, while this is necessary to recognise easily interesting patterns.

This project contributes to both process mining and data analytics. It creates a discovery algorithm suitable for EDA. The resulting models only represent the observed behaviour and are optimised for comprehensibility. Further contributions of this project are a first comprehensibility measure which takes duplicate tasks into account and alternative visualizations for partial parallelism and longterm dependencies.

**Acknowledgments** I would like to thank FWO for my PhD scholarship.

## References

1. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, Berlin Heidelberg (2011)
2. Aalst, W.V.D., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. on Knowl. and Data Eng p. 2004
3. Buijs, J.C.A.M., Dongen, B.F.v., Aalst, W.M.P.v.d.: A genetic algorithm for discovering process trees. In: 2012 IEEE Congress on Evolutionary Computation. pp. 1–8 (Jun 2012)
4. Dongen, B.F.v., Medeiros, A.K.A.d., Wen, L.: Process Mining: Overview and Outlook of Petri Net Discovery Algorithms. In: Transactions on Petri Nets and Other Models of Concurrency II, pp. 225–242. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2009)
5. Figl, K.: Comprehension of Procedural Visual Business Process Models: A Literature Review. Business & Information Systems Engineering 59(1), 41–67 (Feb 2017)
6. Figl, K., Mendling, J., Strembeck, M.: The influence of notational deficiencies on process model comprehension. Journal of the Association for Information Systems 14(6), 312–338 (2013)
7. Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust process discovery with artificial negative events. Journal of Machine Learning Research 10(Jun), 1305–1340 (2009)
8. Johannesson, P., Perjons, E.: An Introduction to Design Science. Springer (Oct 2014)
9. Jouck, T., Depaire, B.: Generating Artificial Data for Empirical Analysis of Control-flow Discovery Algorithms: A Process Tree and Log Generator. Business & Information Systems Engineering pp. 1–18 (Mar 2018)
10. Kalenkova, A.A., van der Aalst, W.M.P., Lomazova, I.A., Rubin, V.A.: Process mining using BPMN: relating event logs and process models. Software & Systems Modeling 16(4), 1019–1048 (Oct 2017)
11. Kunze, M., Berger, P., Weske, M., Lohmann, N., Moser, S.: BPM Academic Initiative-Fostering Empirical Research. In: BPM (Demos). pp. 1–5 (2012)
12. Muñoz-Gama, J., Carmona, J.: A Fresh Look at Precision in Process Conformance. In: Business Process Management. pp. 211–226. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (Sep 2010)
13. La Rosa, M., Wohed, P., Mendling, J., Ter Hofstede, A.H., Reijers, H.A., van der Aalst, W.M.: Managing process model complexity via abstract syntax modifications. IEEE Transactions on Industrial Informatics 7(4), 614–629 (2011)
14. Lorenz, R., Mauser, S., Juhas, G.: How to synthesize nets from languages - a survey. In: 2007 Winter Simulation Conference. pp. 637–647 (Dec 2007)
15. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: Handbook of metaheuristics, pp. 320–353. Springer (2003)
16. Shneiderman, B., Plaisant, C.: Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In: Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization. pp. 1–7. ACM (2006)
17. Tukey, J.W.: Exploratory data analysis, vol. 2. Reading, Massachusetts (1977)
18. Weijters, A., Aalst, W.M.P., A K Medeiros, A.: Process Mining with the Heuristics Miner-algorithm, vol. 166 (01 2006)