Heuristic Mining Revamped: An Interactive, Data-aware, and Conformance-aware Miner

Felix Mannhardt[1], Massimiliano de Leoni[1] , Hajo A. Reijers[2,1]

[1] Eindhoven University of Technology, Eindhoven, The Netherlands
[2] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
{f.mannhardt, m.d.leoni, h.a.reijers}@tue.nl

Abstract. Process discovery methods automatically infer process models based on events logs that are recorded by information systems. Several heuristic process discovery methods have been proposed to cope with less structured processes and the presence of noise in the event log. However, (1) a large parameter space needs to be explored, (2) several of the many available heuristics can be chosen from, (3) data attributes are not used for discovery, (4) discovered models are not visualized as described in literature, and (5) existing tools do not give reliable quality diagnostics for discovered models. We present the interactive Data-aware Heuristics Miner (iDHM), a modular tool that attempts to address those five issues. The iDHM enables quick interactive exploration of the parameter space and several heuristics. It uses data attributes to improve the discovery procedure and provides built-in conformance checking to get direct feedback on the quality of the model. It is the first tool that visualizes models using the concise Causal Net (C-Net) notation. We provide a walk-through of the iDHM by applying it to a large event log with hospital billing information.

Keywords: Process Discovery · Heuristics Miner · Data-aware Processes · Interactive

1 Heuristic Process Discovery

Data on the execution of processes is recorded by information systems that are used to support processes, i.e., events and contextual data are stored for the execution of activities. Process discovery methods use this recorded execution data (i.e., event logs) to automatically infer a process model. Both less structured processes and the presence of noise in the event log (i.e., out-of-order events, missing events, infrequent variants etc.) are challenging problems for process discovery methods [1].

Several heuristic process discovery methods have been proposed to cope with these challenges and provide insights into processes at the desired level of detail [1]. Most heuristic methods are based on frequency-based measures that determine the strength of dependencies between any two activities based on the observations in the event log. Examples are the Fuzzy Miner (FM) [2], the Flexible Heuristics Miner (FHM) [3], Fodina Miner (FM) [4], and the Data-aware Heuristic Miner [5]. Moreover, most commercial process mining tools (e.g., Fluxicon, Lexmark, Celonis) rely on some form of heuristic discovery. Thus, heuristic process discovery methods are widely used.

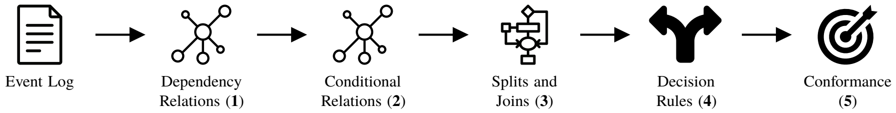


Fig. 1. An overview of the five steps of heuristic process discovery supported by the iDHM.

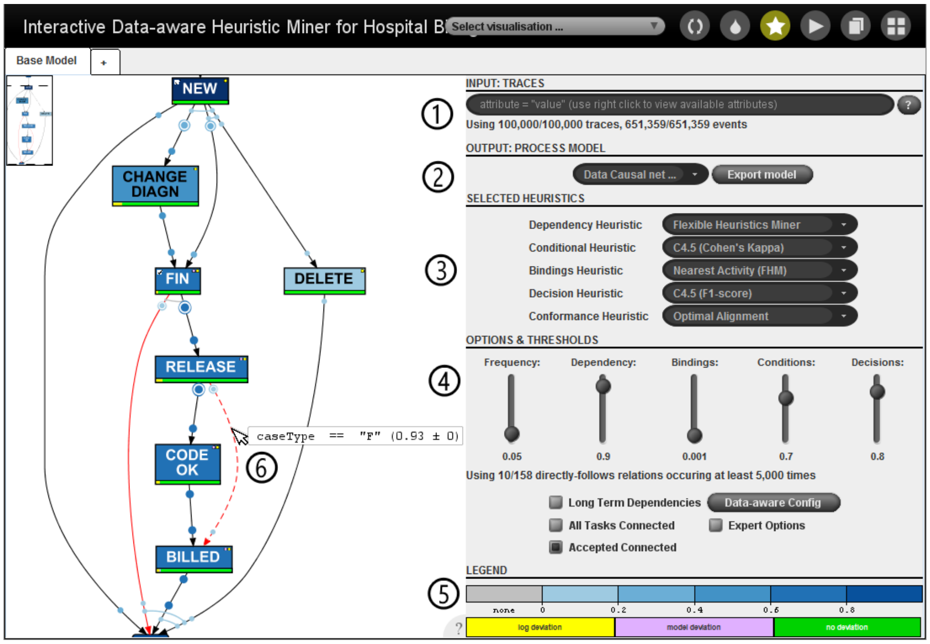
This paper reports on a number of improvements over shortcomings of existing heuristic-based process-discovery tools: (1) the large parameter space of heuristic methods needs to be explored manually; (2) several of the many available heuristics can be chosen from; (3) data attributes (i.e., the event payload) are not used for process discovery; (4) discovered Causal Nets (C-Nets) are not visualized as described in literature; and (5) existing tools do not give reliable quality diagnostics for the discovered models despite the lack of quality guarantees offered by heuristic approaches.
We present the interactive *Data-aware Heuristics Miner (iDHM)*, an open-source tool in the ProM framework[3]. The iDHM aims to help tackling the described issues by integrating several existing methods in a user-friendly tool. The iDHM provides an *interactive exploration* of the parameter space and includes built-in conformance checking to diagnose the quality of the discovered model. Thus, it is *easier to explore a large parameter space* and directly spot conformance issues of the discovered model, e.g., deviating and missing behavior in the event log. Furthermore, the iDHM uses data attributes of the event log to reveal infrequent conditional dependencies. In [5], we show that this helps uncovering potentially interesting process behavior while still filtering random noise. The discovered models are, then, visualized as *Causal Nets (C-Nets)*, a concise graphical notation [1] with *clear semantics*, which includes information on split and join gateways. Existing tools rely on tabular descriptions of the splits and joins (e.g., FHM and FM), i.e., the *exact semantics are not directly visible* in the process model. Finally, the iDHM provides a plug-in architecture that allows for new heuristics to be added.

We applied the iDHM in a case study on an event log with hospital billing information [6]. In the remainder of this paper, we provide a walk-through of the iDHM using this event log. We distinguish five steps of data-aware process discovery (Fig. 1): (1) mining dependencies, (2) mining data-aware conditional dependencies, (3) mining split- and join information, (4) discovering decision rules, and (5) checking conformance. Each step can be configured with heuristics implemented as plug-ins to the iDHM.

2 Interactive Data-aware Heuristic Miner

We present a walk-through of the iDHM using an event log obtained from the billing process of a regional hospital in The Netherlands. Additional documentation and a screencast of this walk-through are available at: `https://fmannhardt.de/g/dhm`.
Figure 2 shows the main screen of the iDHM. The required input is an event log in the XES format. Traces from the event log may be filtered by using a SQL-like query syntax ①. The figures in this paper show the output in form of a C-Net, but other output formats

[3] Available in the *DataAwareCNetMiner* package of ProM 6.7: `http://promtools.org`



INPUT: TRACES

attribute = "value" (use right click to view available attributes)

Using 100,000/100,000 traces, 651,359/651,359 events

OUTPUT: PROCESS MODEL

SELECTED HEURISTICS

Dependency Heuristic	Flexible Heuristics Miner
Conditional Heuristic	C4.5 (Cohen's Kappa)
Bindings Heuristic	Nearest Activity (FHM)
Decision Heuristic	C4.5 (F1-score)
Conformance Heuristic	Optimal Alignment

OPTIONS & THRESHOLDS

Frequency:	Dependency:	Bindings:	Conditions:	Decisions:
0.05	0.9	0.001	0.7	0.8

Using 10/158 directly-follows relations occuring at least 5,000 times

□ Long Term Dependencies
□ All Tasks Connected
□ Expert Options
☒ Accepted Connected

```
caseType == "F" (0.93 ± 0)
```

Fig. 2. The main screen of the iDHM showing a discovered C-Net and conformance information.

such as Petri net or Data Petri net can also be chosen ②. The heuristic used for each step of the iDHM can be configured ③. Several thresholds are used to configure the heuristics ④. At the bottom of the screen a legend explains the used color coding ⑤. Edges can be selected to show more information ⑥.

(Step 1) Discover Dependency Relations. First, the set of dependency relations is determined. A dependency relation (a, b) between two activities represents a causal dependency from activity **a** to activity **b**, e.g., in Fig. 2 the activity DELETE depends on a prior execution of the activity NEW. This is depicted as a directed edge between both activities. Only strong dependencies that exceed a configured *dependency* threshold and are observed more often than a configured observation *frequency* threshold are included. Several heuristics have been proposed to discover dependency relations and their strength. We allow four choices: the FHM [3], the Alpha Miner [1], the FM [2], and a combination of these three miners by taking the average dependency value obtained. In Fig. 2, we used the FHM method. New methods can be added as plug-ins to the iDHM.

(Step 2) Discover Conditional Dependency Relations. Second, the data perspective is taken into account when discovering the control flow of a process. Classification techniques are used to reveal data dependencies between activities. Those data dependencies are used to distinguish random noise from infrequent conditional dependencies as described in [5]. This greatly extends existing techniques, which are solely based on the control-flow perspective and, hence, would disregard such infrequent behavior as noise.

Conditional dependencies may provide insights for process analysts since they could indicate, e.g., workarounds and deviations from normal process behavior. In Fig. 2, e.g., the dependency between FIN and the end of the process ⑥ is a conditional dependency that was discovered based on the data condition $closeCode = H$ with a quality score of 0.98. Conditional edges are highlighted in red. Further statistics on the discovered condition can be obtained by right-clicking on the edge. We only include conditional dependencies for which the quality of the underlying data condition exceeds the configured condition threshold. We implemented two plug-ins, one that uses Cohen's kappa as proposed in [5] and one that uses F1-score as quality measure.

(Step 3) Discover Split- and Join Information Third, the split- and join information of the C-Net, i.e., its input- and output bindings, need to be discovered to obtain a model with precise semantics. Bindings are visualized as dots on the edges of the C-Net. Disconnected dots represent exclusive splits (XOR) and connected dots represent parallel splits (AND). In Fig. 3, after executing RELEASE an exclusive choice between either BILLED or CODE OK is modeled, i.e., the binding dots are not connected with each other. Currently the heuristic proposed by the FHM in [3] is implemented and only frequent bindings exceeding the *bindings threshold* are displayed. In future work, other heuristics may be added, such as the structuring approach presented in [7].

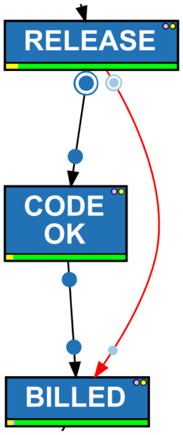


Fig. 3. Binding frequencies, guarded bindings and conformance information.

(Step 4) Discover Decision Rules. Fourth, decision rules that determine which of the output bindings may be activated are discovered. Bindings with attached decision rules, i.e., guarded bindings, are depicted by a double border. In Fig. 3 decision rules could be discovered for both output bindings of RELEASE. The binding from RELEASE to CODE OK is activated only for cases with the *caseType* attribute values A,B, and D. The binding from RELEASE to BILLED is activated only for the values C, F, and G. Additional information on the decision rule can be accessed by right-clicking on the binding. Decision rules may be filtered based on a decision-rule quality threshold. We implemented two decision mining methods, one based on C4.5 decision trees and one based on overlapping decision rules [8].

(Step 5) Check Conformance of the Model. Finally, we apply conformance checking techniques on the C-Net and the event log to project frequencies on the activities and bindings. Activities and bindings are color-coded based on their frequency of occurrence according to the event log. The size of the binding dots also scales with their frequency of occurrence. In Fig. 3 the binding from RELEASE to CODE OK occurs much more often than the binding from RELEASE to BILLED. Moreover, conformance problems are diagnosed: some events regarding the activity BILLED are not represented in the model (yellow bar) and very few executions of BILLED are not observed in the log (purple bar). Additionally, we add a small circle of the respective color to the top left of the activity to indicate the presence of conformance issues. Several heuristic and exact approaches are possible. Currently, we implement both the nearest activity heuristic introduced by the FHM [3] and an optimal

multi-perspective alignment [9] based on a Data Petri net representation of the C-Net. A possible addition in future work would be, e.g., the heuristic execution semantics proposed in [4] or approaches that approximate an optimal alignments to provide faster feedback.

3 Conclusion

We presented the iDHM, an interactive data-aware heuristic process discovery tool that is integrated with the ProM framework. The tool integrates several available heuristics and is envisioned as platform for future research on new heuristics. We showed how the iDHM improves upon several shortcomings of academic heuristic process discovery tools. Its modularity and interactive UI allow for a quick exploration of the parameter space (1) and the available heuristics (2). It makes use of data attributes to improve the discovery of the control-flow (3). The result is visualized as C-Nets with clear semantics (4). Finally, built-in conformance checking directly reveals quality problems of the models discovered (5). We successfully tested the iDHM on several real-life event logs with up to 6 million events, which confirms that it has reached a high degree of maturity. In this paper, we applied it to an event log of a hospital billing process with more than 450,000 events [6]. A limitation is the dependence on an automatic layout based on the GraphViz software. Very complex C-Nets are unlikely to be readable and it is not yet possible to manually adjust their layout. As future work we plan to further improve its usability and enrich the C-Net notation with perspectives different than control-flow. We would also like to investigate the understandability of C-Nets among users and aim to add additional heuristics as plug-ins.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Günther, C.W., van der Aalst, W.M.P.: Fuzzy mining - adaptive process simplification based on multi-perspective metrics. In: BPM 2007. Volume 4714 of LNCS., Springer (2007) 328–343
3. Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible Heuristics Miner (FHM). In: CIDM 2011, IEEE, IEEE (2011) 310–317
4. vanden Broucke, S.K., Weerdt, J.D.: Fodina: A robust and flexible heuristic process discovery technique. Decis Support Syst (2017) (to appear).
5. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Data-driven process discovery - revealing conditional infrequent behavior from event logs. In: CAiSE 2017. Volume 10253 of LNCS. (2017) 545–560
6. Mannhardt, F.: Hospital Billing - Event Log. Eindhoven University of Technology. Dataset. (2017) `https://doi.org/10.4121/uuid:76c46b83-c930-4798-a1c9-4be94dfeb741`.
7. Augusto, A., Conforti, R., Dumas, M., Rosa, M.L., Bruno, G.: Automated discovery of structured process models: Discover structured vs. discover and structure. In: ER 2016. Volume 9974 of LNCS. (2016) 313–329
8. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Decision mining revisited - discovering overlapping rules. In: CAiSE 2016. Volume 9694 of LNCS., Springer (2016) 377–392
9. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Balanced multi-perspective checking of process conformance. Computing **98**(4) (2016) 407–437