

The Multi-perspective Process Explorer

Felix Mannhardt^{1,2}, Massimiliano de Leoni¹, Hajo A. Reijers^{3,1}

¹ Eindhoven University of Technology, Eindhoven, The Netherlands

² Lexmark Enterprise Software, Naarden, The Netherlands

³ VU University Amsterdam, Amsterdam, The Netherlands

{f.mannhardt, m.d.leoni, h.a.reijers}@tue.nl

Abstract. Organizations use process mining techniques to analyze event data recorded by their information systems. Multi-perspective process mining techniques make use of data attributes attached to events to analyze processes from multiple perspectives. Applying those multi-perspective process mining techniques in practice is a laborious task when the event data contains a large number of attributes and many different trace variants. Tools that facilitate the usage of these techniques in practical settings are missing. We describe the *Multi-perspective Process Explorer* as a new tool that integrates current multi-perspective process mining techniques for discovery and conformance checking. It supports common tasks in multi-perspective process mining, and aims to reduce the time needed to explore event data.

Keywords: Process Mining, Process Analysis, Multi-perspective Process Mining, Interactive Visualization, Process Exploration

1 Multi-perspective Process Mining

Process mining techniques enable organizations to gain insights into their processes by using event data recorded by their information systems. The two major areas of process mining are *process discovery*, i.e., the discovery of process models based on sequences of events, and *process conformance checking*, i.e., revealing differences between recorded executions and behavior prescribed by existing process models [1]. Multi-perspective process mining techniques go beyond techniques that only use event sequences to analyze the control-flow of a process: Data attributes attached to events are used to analyze processes from other perspectives, e.g., discovering and checking decision rules, analyzing resource behavior and checking time-related rules [2,3]. We refer to those techniques as *multi-perspective* as they obtain information about the other perspectives from data encoded in event attributes. Applying multi-perspective process mining techniques in practice is a laborious task, especially in cases when the data contains a large number of different attributes with high variability. A substantial amount of manual work by analysts is required, because they need to filter and transform event data as well as to select relevant features. Also, results need to be explored and, if not satisfactory, these steps need to be repeated multiple times by hand.

Existing commercial tools, such as *Perceptive Process Mining* and *Fluxicon Disco*, and academic tools, such as *Inductive Visual Miner (IVM)* [4], mainly focus on the

Copyright ©2015 for this paper by its authors. Copying permitted for private and academic purposes.

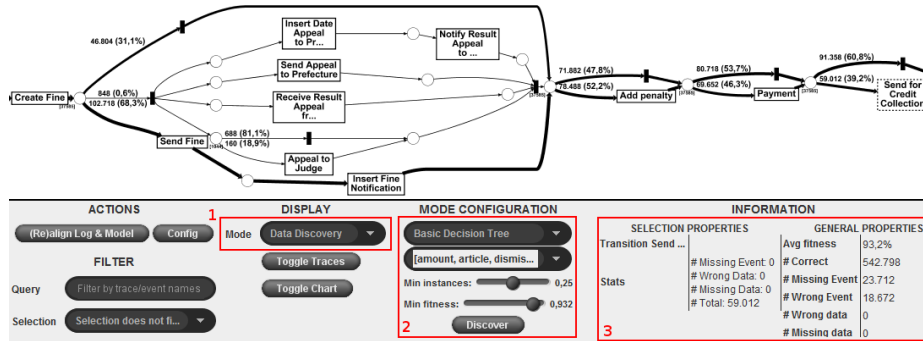


Fig. 1. Screen shot of the MPE where path frequencies are projected onto the model

control-flow perspective, and provide only very limited support for data-aware process exploration. In this paper, we describe the *Multi-perspective Process Explorer* (MPE), a new tool that is tailored towards multi-perspective process exploration for discovery and analysis. It integrates existing work on multi-perspective process mining [2,3] with new interactive visualizations and filtering facilities into a scalable and extensible tool. The main features are: *integration* of existing data-aware discovery, conformance checking, and performance analysis techniques; *interactive* efficient exploration of data-aware processes; built-in *filtering* based on attributes and trace variants. In the reminder of this paper we describe how to use the MPE for one of the possible *use cases*, the discovery of data-aware process models.

2 Walkthrough of The Multi-perspective Process Explorer

The MPE is available as plug-in in the *MultiPerspectiveProcessExplorer* package in the open-source framework ProM¹. Here, we show the sequence of steps required to discover and evaluate a multi-perspective process model. In particular, we showcase the MPE on a publicly available event log containing events from more than 150,000 process instances of a road fines management process in an Italian local police force [3]. An application of the MPE on this event log has also been recorded as a video that is available under <http://purl.tue.nl/899817766269492>.

Required input. Starting point for the usage of the MPE is an event log and a process model, in form of a Petri net. Petri nets can be discovered, e.g., by applying plug-ins implementing several process-discovery techniques. Alternatively, they can be created manually with an editor such as *WoPeD*.² For our case study, we discovered a Petri net using the IVM [4]. Optionally, the model may already contain activity guards. In this case, Steps 1 and 2 can be skipped as they are not necessary. However, one may still want to discover additional activity guards, regardless of those which were already provided.

¹ Available in the ProM nightly builds under: <http://promtools.org>

² <http://woped.org/>

Step 1: Analysis of the Input Model. The first step is to analyze the model provided as input. Figure 1 shows a screenshot of the MPE where the input process model is shown, along with information about the frequencies of paths as observed in the event log. This information is projected onto the model: The thickness of an arc indicates the frequency of observing a path including that arc in the event log. Below the process model, there are three areas highlighted by a red rectangle and marked with a number. The panel in area 3 gives general information about the fitness of the model wrt. the input event log. The fitness score measures whether the behavior observed in the event log is reflected in the model; For the model and the event log in question, the fitness score is 93.2%. This is computed by constructing an alignment between the model and the event log [3], which makes it possible to pinpoint the deviations that cause nonconformity. An alignment between a recorded process execution and a process model is a pairwise matching between activities recorded in the log and activities allowed by the model. Sometimes, activities as recorded in the event log (events) cannot be matched with any of the activities allowed by the model (process activities), thus resulting in so-called *moves on log*. In other cases, an activity should have been executed but is not observed in the event log, thus resulting in a so-called *move on model*. For our case study, there are 23,712 model moves, i.e. missing executions of activities, and 18,672 log moves, i.e. activity executions that occurred when not allowed by the model.

Step 2: Data-aware Discovery. This step is about discovering the guards associated with process activities. To do that, we need to switch the mode to *data discovery* in the *Display* panel (area 1 in Fig. 1). In the data discovery mode, the *mode configuration* panel looks like area 2 in Fig. 1. It allows users to configure which guard-discovery algorithm to use, which data attributes to consider along with tuning some specific algorithms parameters: the minimum number of elements associated with decision-tree leaves (*min instances*) and the minimum control-flow fitness for each trace to be considered (*min fitness*). For further information, readers can refer to [2]. Once parameters are chosen, the button *Discover* can be pressed. In this case, we choose the standard decision tree classifier configured to use all attributes and 25% as the *min instances* parameter, i.e., the number of elements is set to 25% of the process instances reaching a decision point. The *min instances* parameter is important as it influences whether the discovered guards are over-fitting (value too low), or under-fitting (value too high).

Step 3: Fitness & Precision Computation. After discovering a multi-perspective process model, the plug-in can evaluate the quality of the discovered process model. This requires to change the mode of the MPE to *fitness* in the *Display* panel mentioned above. Please observe that there is a substantial difference compared with Step 1: now the model contains activity guards. Fig-

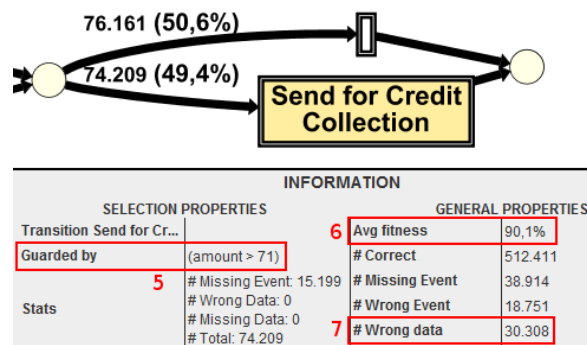


Fig. 2. An excerpt of the visualization when in *fitness* mode.

Figure 2 shows an excerpt of the visualization when in *fitness* mode. In particular, the focus is on the activity *Send for Credit Collection*, for which the following decision rule has been discovered: $\text{amount} > 71$ (area 5). Each activity is colored according to the ratio between the number of compliant executions of the activity and the total moves for that activities (which also accounts for missing events and executions with incorrect data). The relatively dark color of the *Send for Credit Collection* activity indicates a considerable fraction of non-compliant executions. Moreover the statistics show that according to the new model the data values observed in the event log are wrong 30,308 times (area 7). If we compare the average fitness of the model with and without activity guards (compare area 3 in Fig. 1 with area 6 in Fig. 2), it is clear that the presence of data guards has decreased the fitness level from 93.2% to 90.1%. However, this is not necessarily negative: fitness is only one of the measure to evaluate the quality of a model. A second measure is *precision*, which is the ratio between the amount of behavior observed in the event log and the amount of behavior described by the model. Adding guards increase the precision because the added rules restrict the behavior allowed by the model. The MPE also allows for computing precision [5]. This is done by switching the display mode to *precision*. For sake of space, a detailed description of the *precision mode* is only showcased in the screencast.

Step 3: Bottleneck & Performance Computation After evaluating whether the discovered model is a suitable representation of the process behavior, performance information about the time perspective such as average waiting times can be obtained using the *performance mode* of the MPE. Figure 3 shows that the average waiting time between the activity *Create Fine* and the activity *Send Fine* is 7.4 hours (area 8). In this case, we have also used a *filtering feature*: $\text{amount} > 71$ (area 9). This filtering query restricts the analysis to only those traces for which attribute *amount* is larger than 71 at least once in the trace.

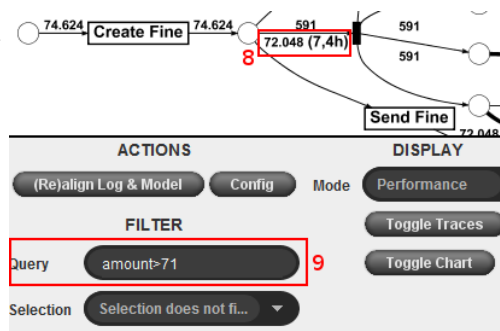


Fig. 3. An excerpt of the *performance mode* screen. This filtering query restricts the analysis to only those traces for which attribute *amount* is larger than 71 at least once in the trace.

Step 4: Detailed Analysis using Trace View & Chart View. After evaluating the fitness and precision of the model, and the presence of bottlenecks, the end user may want to explore specific traces in detail that she, e.g., finds to be problematic. Therefore, the MPE provides two complementary views on the process showing more details. Figure 4a shows the detailed *trace view* that opens on a second screen upon pressing the *Toggle Traces* button. For each log trace, the corresponding alignment is shown: log and model moves are highlighted with yellow and purple color above the move, wrong data is highlighted with white color. Moves related to the same activity are painted with the same color. Trace are grouped based on similar executions. Individual traces and their data attributes can also be explored. The second view, shown in Figure 4b, provides more details on the distribution of data attributes at certain states within the process model. Figure 4b shows two histograms with the distribution of the values of attribute *amount* before the occurrence of activity *Send for Credit Collection* and

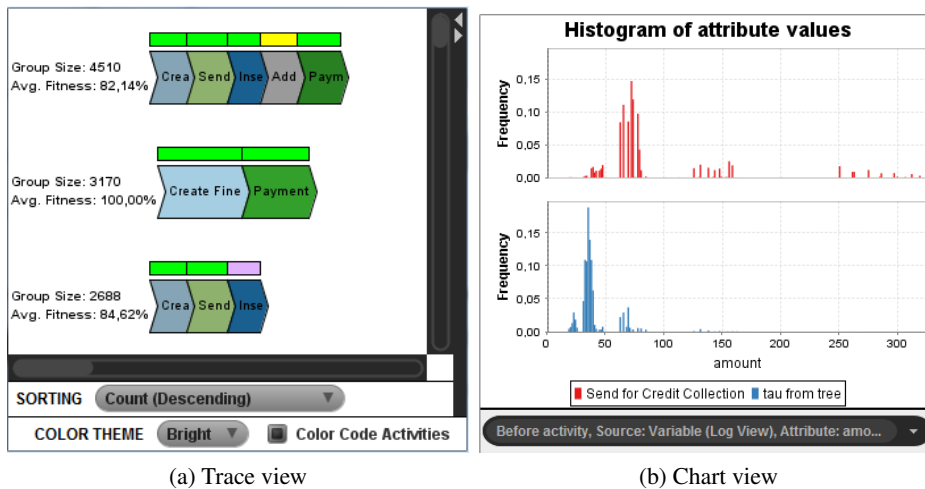


Fig. 4. Two additional views for detailed analysis

the invisible step *tau from tree*, which models the skipping of *Send for Credit Collection*. This allows end users to visually analyze whether certain ranges of values are usually observed together with the occurrence of given activities. For instance, for *Send for Credit Collection*, the observed values are usually high.

Conclusion. We presented the MPE as a novel tool for multi-perspective process exploration. For the sake of space, we only showed one iteration cycle. However, any of the steps can be repeated as many times as necessary. The tool has reached a high degree of maturity, which allows it to be used for real-life case studies. This is testified by its application on the analysis of a real-life event log with more than 150,000 traces and around 500,000 events. As next steps, we plan to evaluate the user interface with real process analysts and improve it when necessary. Also, we are currently working on novel data-discovery techniques, which we aim to incorporate by the end of 2015. Finally, we aim to work on improving the performance of the used alignment techniques to further speed up the analysis of large data sets and complex process models.

References

1. van der Aalst, W.M.P.: *Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer (2011)
2. de Leoni, M., van der Aalst, W.M.P.: *Data-Aware Process Mining: Discovering Decisions in Processes Using Alignments*. In: SAC'13, ACM (2013) 1454–1461
3. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: *Balanced multi-perspective checking of process conformance*. *Computing* (2015) doi:10.1007/s00607-015-0441-1 (in press).
4. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: *Process and deviation exploration with inductive visual miner*. In: *Proceedings of the BPM Demo Sessions 2014*. Volume 1295 of *CEUR Workshop Proceedings.*, CEUR-WS.org (2014) 46
5. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: *Measuring the Precision of Multi-perspective Process Models*. In: *BPI'15*. (2015) (accepted).