

Directly Follows-Based Process Mining: a Tool

Sander J.J. Leemans, Erik Poppe, Moe T. Wynn
Queensland University of Technology
Brisbane, Australia
{s.leemans, e.poppe, m.wynn}@qut.edu.au

Abstract—In order to bridge the gap between well-founded academic and intuitive commercial tools, we introduce the **Directly Follows visual Miner (DFvM)**, which takes as input an event log and allows users to explore it. That is, it automatically discovers a directly follows process model, applies conformance checking, provides performance measures and allows for filtering the log. DFvM uses directly follows models, which are also used by many commercial process mining tools, however, unlike such tools, it provides conformance checking and reliable performance measures.

Index Terms—process discovery, conformance checking, process enhancement, performance mining, directly follows-based process mining

I. INTRODUCTION

Process mining aims to obtain insights from event logs that contain recorded behaviour of executions of an organisation's business processes, in order to optimise the processes. Typically, in a process mining project, first a process model that describes the control flow of the process is discovered from an event log. Second, the model should be evaluated against the event log or a secondary test event log to verify that the model represents the behaviour of the process well. This is typically performed using a conformance checking technique. Third, the performance of the process can be measured to identify bottlenecks, central concepts and batching behaviour. These steps can be repeated to, for instance, zoom in on areas of particular interest, or to compare different groups of recorded behaviour [1].

Many process mining techniques and tools have been proposed, both commercially (e.g. [2]) and academically [3]. Academic tools typically use process models with a well-defined semantics that support advanced constructs such as concurrency, interleaving and inclusive choices. This allows such models, for instance Petri nets, process trees or BPMN models, to be evaluated using conformance checking techniques [4]. In contrast, many commercial process mining tools use directly follows-based process models to convey business processes to stakeholders, even though these models often lack an executable semantics and support neither concurrency, interleaving nor inclusive choices. As the executable semantics of such models is not clear, obtained insights are difficult to verify [4]. Nevertheless, such models are considered to be easier to understand for users [3].

In [4], we introduced concepts to bridge this gap between academic and commercial process mining tools: we keep the directly follows based models, but use a proper semantics, introduce soundness and describe how conformance checking

and performance measuring concepts can be applied to these models.

In this paper, we present the *Directly Follows visual Miner* (DFvM), which applies both well-established and new academic concepts to directly follows based models. Our aim is to illustrate that conformance checking and reliable performance measures are possible within the limitations of intuitive directly follows-based models.

Relation with Inductive visual Miner: The DFvM is an extension of the Inductive visual Miner (IvM) [1] and shares its code base. Compared to the previously published version of IvM [1], DFvM has a new architecture (Section III) and several new features (Section IV). The architecture and many of the new features of DFvM have been made available to users of IvM as well, however have not been published before.

In the remainder of this paper, we first introduce the idea of the DFvM (Section II) and explain its architecture (Section III). We describe the new features in Section IV and describe its maturity in Section V. Section VI explains how the tool can be accessed while the paper is concluded in Section VII.

II. IDEA

The Directly Follows visual Miner (DFvM) takes an event log and automatically applies a series of steps, which allows the user to perform process-based analyses on the log. That is, first a Directly Follows Model (DFM) is discovered by a DFM discovery algorithm. Second, the DFM is aligned with the event log, and the results are shown. That is, deviations between event log and DFM can be shown on both the model (to show where deviations occur) and on the event log (to show which events do not correspond to the model [log moves], and where in the traces the model made a move that is not represented in the trace [model moves]). Furthermore, several performance measures, such as how often parts of the model are executed and the time spent waiting for or executing these parts, are computed and visualised. In addition, the event log is animated as yellow tokens flowing through the model, which allows the observation of bottlenecks, batching behaviour and seasonality. In typical process mining projects, insights gained using these techniques might highlight the need for drilling further into the event log, for instance to compare different subsets of the event log or to zoom in on a particular part of the DFM [1]. DFvM facilitates such inquiries by quick filtering in several ways: first, clicking an activity or edge of the DFM, filters the log to contain only traces that use that activity or

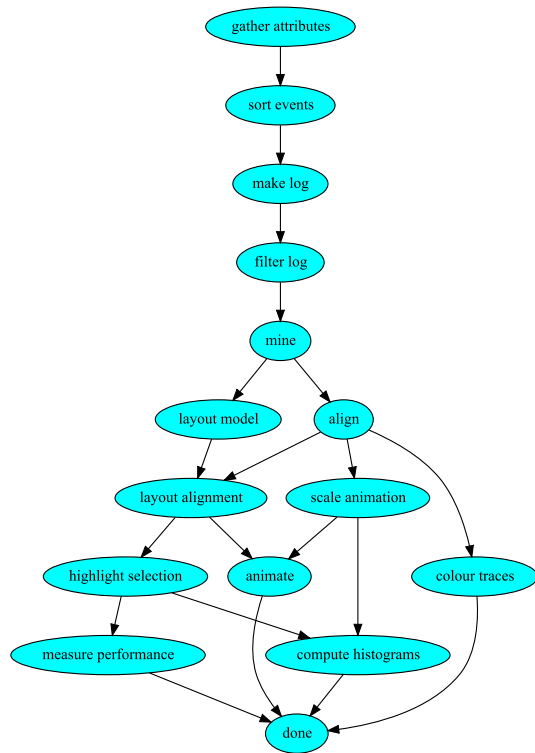


Fig. 1. Architecture of the Directly Follows visual Miner.

edge. Second, filters based on trace and event attributes can be applied before or after discovery of the DFM. Any such filtering will update all performance measures accordingly.

To summarise, using DFvM a user can *explore* the process iteratively to derive insights by repeatedly discovering a process model using process discovery, evaluating the model using conformance checking, assessing its performance and drilling down by filtering the log.

To enable this seamless exploration, a new architecture underlying DFvM automates this process: when users change a setting DFvM recomputes results as necessary.

III. ARCHITECTURE

DFvM performs several steps fully automatically: after a few log-related steps (filtering, applying the classifier, etc.), a directly follows model is discovered, after which it is aligned, the log is filtered and the results of that filtered alignment are visualised (performance, histograms, animation, colouring, etc.).

The architecture of the DFvM is shown in Figure 1: the graph shows the steps that are performed to visualise event logs and directly follows models, and the dependencies between steps. Each step is executed (in parallel) as soon as all preceding steps (that is, all steps with arcs going into the current step) have been completed. Similarly, when a user changes a setting, only the relevant dependent steps are recomputed. For instance, if a user changes the highlighting selection (e.g. by clicking on an activity to filter the log to only include traces going through that activity according to the model) then

highlight selection, measure performance and compute histograms are redone. This architecture is supported by a flexible multi-threaded framework that uses this graph to take care of starting steps, keeping track of their progress, and cancelling steps and voiding their results if they are no longer necessary (because the user changed something upstream again).

This framework has proven to be flexible: new steps can be added to the framework by simply adding them and their dependency edges to the graph.

Compared to IvM, the framework has been improved: previously, all current steps were executed sequentially and only if a step's result would become obsolete a concurrent execution was started.

IV. MAIN NEW FEATURES

Directly Follows visual Miner (DFvM) is based on the Inductive visual Miner (IvM) [1] and extends it as follows with new features exclusively for directly follows models:

- **Process discovery.** The new process discovery algorithm is described in [4] and is available in DFvM. This algorithm simplifies the model by filtering traces, based on the paths slider. Using the paths slider, the level of infrequent behaviour filtering can be adjusted. The value of the slider sets the minimum percentage of traces in the event log that fit the model (guaranteed when only completion events are considered [4]).
- **Conformance checking.** Directly follows models are automatically translated to Petri nets and aligned using [5]. The results are used throughout the DFvM, for instance to show deviations, animate the event log over the model and to compute performance measures. DFvM shows deviations using the concepts introduced in [4], which illustrates how (commercial) directly follows-based tools could implement conformance checking.
- **Performance.** Typically, existing DFM-based tools compute performance measures based on DFM-edge traversal. As shown in [4], this might lead to counter-intuitive results. DFvM, on the other hand, computes performance measures based on the computed alignment. That is, the traces are projected on the model and several performance measures are computed based on this projection: minimum, average and maximum sojourn, waiting and service time, and elapsed and remaining trace time.
- **Edit model.** Users can manually edit the directly follows model that was discovered, or import an existing stored model to avoid discovery altogether. Editing the model may involve changing the start and end activities, as well as the edges that make up the model and selecting whether the model should support empty traces. While editing the model, users are shown an example of the to-be-model to help with editing and creating a *sound* model, that is, a model in which every activity is reachable and can proceed to the end of the model [4]. If the model is not sound, DFvM will highlight the identified issue (see Figure 3). As soon as the edited model is sound, DFvM

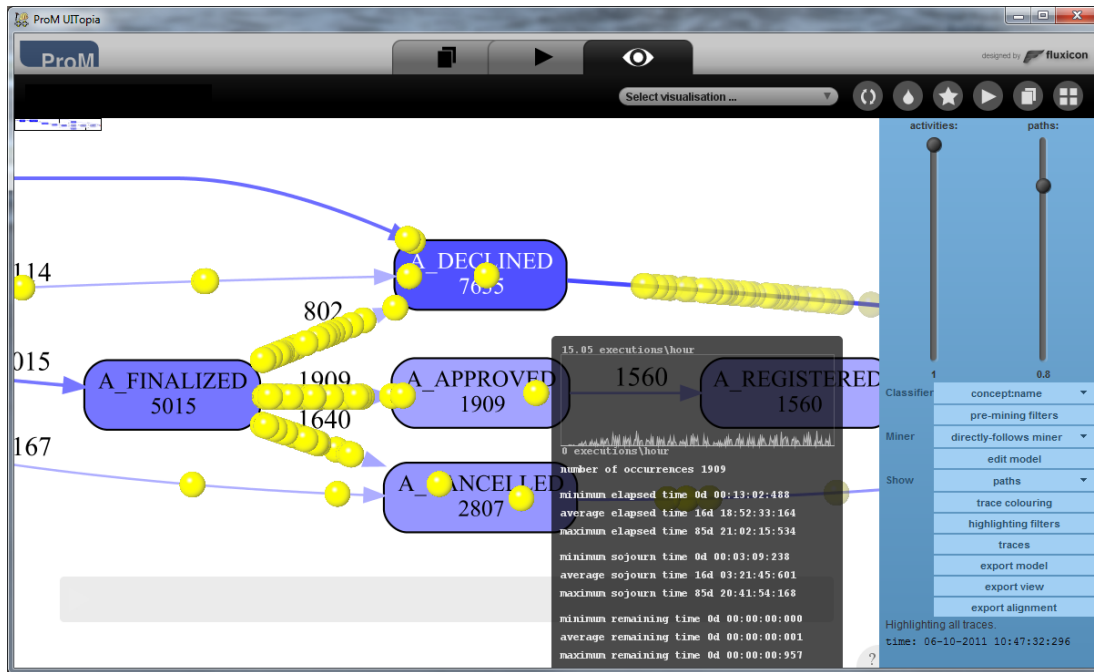


Fig. 2. A screenshot of the Directly Follows visual Miner, showing the controls on the right, the model on the left and the performance measures of an activity in a pop-up.

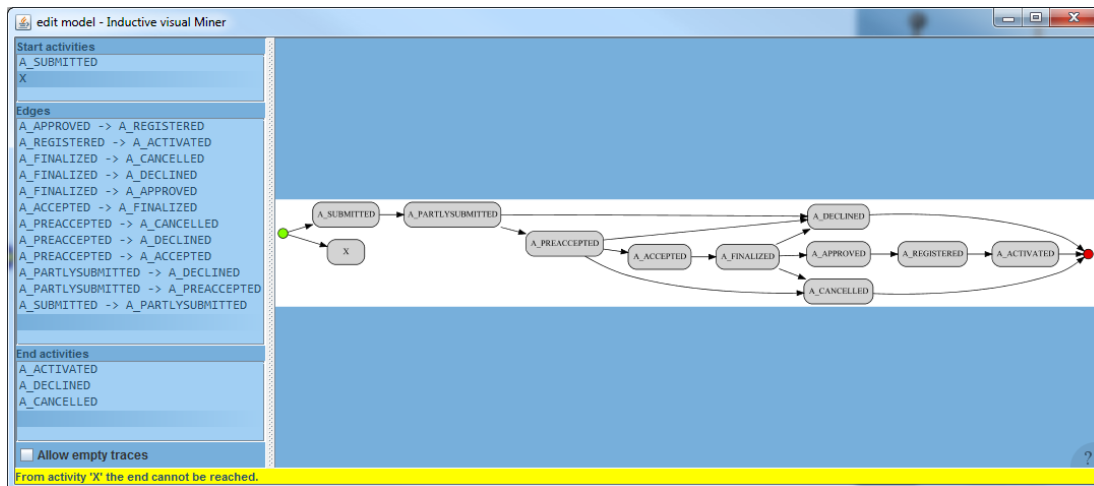


Fig. 3. A screenshot of DFvM's model editing capabilities. A sound model is necessary for further processing. The shown model is not sound as from activity X, the end state cannot be reached. A preview of the model is shown in order to aid users to correct the issue.

will adopt the new model and continue computations according to Figure 1.

- Export model. The directly follows model can be exported to ProM as a Petri net, an Accepting Petri net, an Expanded Accepting Petri net (with explicit start and completion activities) or as a DFM.

New features of DFvM that also benefit users of IvM:

- Trace and token colouring. Using “trace colouring”, a trace-level attribute can be chosen, and every trace will be coloured according to the value of this attribute using a continuous Viridis colour map. The tokens, as well as the

traces in the trace view, are highlighted with this colour.

- Hardware accelerated drawing. The drawing of tokens now makes use of hardware acceleration using OpenGL for smoother animations.
- Edit model for process trees. A model editing facility has been added for process trees as well.
- Additional performance measures: DFvM shows elapsed and remaining time, in addition to waiting, service and sojourn time. For an activity a , *elapsed time* is the time from the first timestamp in a trace to the start of the execution of a , or the completion if no start timestamp is available. Similarly, *remaining time* is the time of

the completion (or start if that is not available) of a to the last timestamp in a trace. The reported measures are the minimum, average and maximum times over all executions of an activity a .

- Import an existing process model and bypass the discovery. Conformance checking and performance analysis for existing models can be done using the ProM plug-in “Visualise deviations on directly follows model (Directly Follows visual Miner)”.
- Switch between DFvM and IvM. Users can easily switch between DFvM and IvM by choosing a different “miner”: even though DFvM and IvM use completely different concepts, if advanced process discovery using process trees with concurrency, interleaving or inclusive choices is necessary, IvM is two clicks away.

V. MATURITY

The Inductive visual Miner was originally introduced in 2014, and has attracted 53 [1] and 41 [6] citations (Google Scholar, 18-03-2019) and has been used in several industry projects within QUT’s BPM discipline.

We have applied the Directly Follows visual Miner (DFvM) to a case study in a Queensland Government department [4], in which it proved useful to perform process mining analyses on models that were difficult to represent with Petri nets or process trees. Furthermore, the department had 72 manually created directly follows-based business process descriptions, which were transformed to a DFvM format, after which we used DFvM to apply conformance checking to these models and corresponding event logs to gain insight into deviations, frequencies and performance [4].

Finally, the new edit model feature has been used in collaborations with several industry partners during workshops. While discussing the model with the participants, it is modelled live in DFvM, after which DFvM quickly shows the alignment results (deviations, frequencies and performance), which then often leads to further discussions on the model and on the process. For instance, one of the participants of a workshop noted “Utilising the DFvM tool during workshops has been beneficial. Being able to visually see the deviations, frequencies and performance of the processes has facilitated robust discussion. All workshop participants, whether process focussed or not, are able to quickly comprehend the model and leverage the findings to identify process inefficiencies and drive potential process transformation.” (Janne Barnes, Project Manager - Queensland University of Technology - Research Management Systems Upgrade).

VI. ACCESS & DEMONSTRATION

The Directly Follows visual Miner (DFvM) is available both as a ProM plug-in (<http://promtools.org>, from version 6.9 or in Nightly Builds) and as a part of the ProM QuickVisualiser (<http://leemans.ch/quickvisualiser/>).

In both cases, an event log is required to run DFvM. Example event logs are available from https://data.4tu.nl/repository/collection:event_logs_real.

A user manual is available from <http://leemans.ch/inductivevisualminer/>, and a screencast video is available on <https://youtu.be/xTKKqGwzh6I>.

VII. CONCLUSION

Typical process mining projects involve repeated process discovery, conformance checking, filtering and performance measuring. Academic process discovery techniques might return models that are difficult to understand or that under- or overfit the event log, while commercial tools typically do not offer conformance checking capabilities (and thus, might provide counter-intuitive performance measures).

In this paper, we introduced the Directly Follows visual Miner (DFvM): a new tool that automatically discovers a directly follows model, applies conformance checking (alignments), computes performance measures and supports filtering to drill down into parts of the event log.

DFvM extends and improves the Inductive visual Miner with a new formalism (directly follows models rather than process trees), a new more concurrent architecture using a flexible framework, the ability to edit models and new performance measures.

We express the hope that the ideas illustrated in DFvM will lower the bar for analysts to apply conformance checking and that it will inspire commercial vendors and others to include conformance checking capabilities.

REFERENCES

- [1] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Process and deviation exploration with Inductive visual Miner,” in *BPM Demo Sessions 2014*, 2014, p. 46.
- [2] C. W. Günther and A. Rozinat, “Disco: Discover your processes,” in *BPM demos*, 2012, pp. 40–44.
- [3] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.
- [4] S. J. J. Leemans, E. Poppe, and M. T. Wynn, “Directly follows-based process mining: Exploration & a case study,” in *ICPM*, 2019, in print.
- [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,” *WIRDMKD*, vol. 2, no. 2, pp. 182–192, 2012.
- [6] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, “Exploring processes and deviations,” in *BPM Workshops*, 2014, pp. 304–316.