

The Process Mining ToolKit (PMTK): Enabling Advanced Process Mining in an Integrated Fashion (Extended Abstract)

Alessandro Berti*, Chiao-Yun Li†, Daniel Schuster‡, and Sebastiaan J. van Zelst (✉)§
Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany
Process and Data Science (PADS) Chair, RWTH Aachen University, Germany
Email: {alessandro.berti*, chiao-yun.li†, daniel.schuster‡, sebastiaan.van.zelst§}@fit.fraunhofer.de

Abstract—Heaps of event data are being generated and stored during the execution of (business) processes. Over the recent years, various process mining solutions have been developed, i.e., both in industry and academia, that can translate such data into meaningful insights. However, there is a big gap between the number of possible analysis techniques proposed in the literature and the widespread availability of said techniques in commercial applications. At the same time, existing academic tools, i.e., exposing a plethora of analysis techniques, are not designed to be seamlessly integrated into the business nor to provide an end-to-end solution. Therefore, this paper presents the Process Mining ToolKit, i.e., PM_{TK}, intended to bridge the gap mentioned. Building on top of the open-source project PM4Py, PM_{TK} presents novel process mining algorithms and techniques in an easy-to-use, fully integrated solution.

Index Terms—process mining, process analytics, visual analytics, data science

I. INTRODUCTION

The execution of (business) processes generates digital records of historical process behavior, i.e., referred to as *event data*. *Process mining* [1] is concerned with developing techniques and methods that can translate such data into actionable knowledge of the process. Examples of typical process mining techniques include *process discovery*, i.e., automated discovery of process models describing the process based on the event data, and *conformance checking*, i.e., assessing whether the execution of a process as recorded in the event data conforms to a given reference model. Over the recent years, various academic and commercial software solutions have been proposed, implementing process mining technology. Commercial solutions, such as *Celonis* (<http://celonis.com>), *UiPath* (<http://uipath.com>), *Fluxicon Disco* (<http://fluxicon.com/disco/>), etc., often provide basic process discovery functionalities and various (customizable) statistics of the process. Academic tools such as *ProM* [2] (<http://promtools.org>), *Apromore* [3] (<http://apromore.org>), *PM4Py* [4] (<http://pm4py.org>) and *bupar* [5] (<http://bupar.net>) are often open-source and implement a wider range of process mining technologies. Most of these solutions are hard to integrate into a business context or require extensive knowledge of a specific programming language to be used. To bridge this gap, we present the *Process Mining ToolKit* (PM_{TK}). PM_{TK} is built on top of the PM4Py library, i.e., extending our earlier work presented in [6]. As such,

PM_{TK} allows non-technical users to exploit the advanced process mining technology implemented in PM4Py.

II. TOOL OVERVIEW

In this section, we present a short overview of the core components of the PM_{TK} tool. A screen recording corresponding to this extended abstract can be found at <https://pmtk.fit.fraunhofer.de/icpm21/demo.mp4>.¹ We briefly discuss the overall *architecture* of PM_{TK} and its main functionalities, i.e., the *work space*, the main *analysis capabilities* and the *integrated filtering*.

1) *Architecture*: Conceptually, PM_{TK} consists of three different layers: an *algorithmic layer* (based on PM4Py), a *web service layer* (i.e., a controller, based on [6]) and a front-end layer built using web technologies such as HTML5, Javascript and Angular. PM_{TK} is available as a standalone tool, i.e., including the web services and the web interface, and as a web application which can be deployed on any application server.

2) *The Work Space*: PM_{TK} provides a work space in which the user is able to organize various files. Consider Figure 1a, in which we show a snapshot of the work space. In the work space, the user can create a folder for each process she is intending to analyze. Subsequently, various objects, e.g., event logs and filters can be stored in the corresponding process' folder. Some objects, e.g., event logs can be imported from disk, other objects can be generated from within PM_{TK}.

3) *Analysis Capabilities*: When the user selects an object from the work space, various analyses can be applied, based on the selected object. Currently, the user is able to execute the following analyses:

- *Statistics*; PM_{TK} provides various typical event log statistics, i.e., absolute/relative activity occurrences, an overview of the average events per case, events/case arrivals/active cases over time and throughput statistics.
- *Log Exploration*; PM_{TK} provides means to explore the event log in detail, i.e., to gain a better understanding of the process captured by the event log. Currently, PM_{TK} implements the following log exploration functionalities:

¹Based on PM_{TK} release 0.1.1., dated September 2nd 2021. PM_{TK} is available via <http://pmtk.fit.fraunhofer.de>

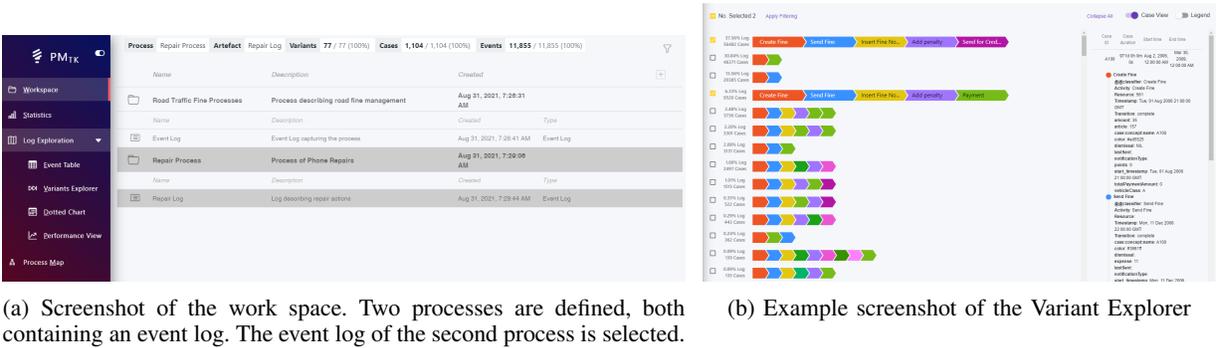


Fig. 1: Screenshots of the workspace and the variant explorer

Variant Explorer: In the variant explorer, the user is able to consult what cases follow the same *control-flow* behavior. See Figure 1b, in which we present a small screenshot of the variant explorer functionality; *Dotted Chart:* PM_{TK} implements the dotted chart analysis, i.e., a visualization of events over time [7]; *Performance View:* PM_{TK} implements the *performance spectrum*, i.e., as described in [8].

- *Process Map;* PM_{TK} implements a process map with various filtering options (i.e., filtering of edges and activities). The layout algorithm implemented is based on [9].

4) *Integrated Filtering:* In PM_{TK}, event data filtering is considered a primary citizen. As such, various event data filtering functionalities have been implemented. The user is able to specify custom filters, e.g., based on start/end activities, time-ranges, etc. Most of the analysis functionalities described in subsection II-3, provide interactive filtering functionality as well. The filters created can be stored in the work space, e.g., to be re-applied at a later stage of the analysis.

III. CONCLUSION

Various software solutions exist that are able to translate recorded event data into operation insights into the historical execution of a process. However, commercial applications only offer a marginal fraction of the algorithmic possibilities, i.e., available in the process mining literature. Academic and open source solutions do provide a larger range of functionalities, yet, often in a non-intuitive manner. In this paper, we have presented the Process Mining ToolKit (PM_{TK}), which aims to bridge this gap, i.e., integrating advanced algorithms in an integrated, user-friendly environment. As such, PM_{TK}, can be seen as a front-end solution for the advanced open source process mining library PM4Py.

Tool Maturity & Novelty: The Fraunhofer FIT process mining team has developed PM_{TK} to provide an extensible, customizable, easy-to-maintain product to its R&D project partners. Compared to [6], the web-service architecture has been redesigned to increase the tool’s modularity. We have adopted object relational mapping for multi-database support, offering support for different artefacts (e.g., the integrated filters) for the same process, i.e., exposed as the *work space*.

Furthermore, all visualizations are now rendered in the front-end and the layout-algorithm of the process map has been redesigned, and, a performance overlay has been added. All log exploration analyses have been added w.r.t. our previous work, i.e., the dotted chart, variant explorer and the performance spectrum.

Future Work: The adoption of new functionalities in PM_{TK} is fairly straightforward, i.e., any algorithm in PM4Py is easily adopted by exposing it as a web-service in the PM4Py-WS service and correspondingly designing a corresponding visualization. As future work, we aim to integrate several new functionalities, e.g., various process discovery algorithms, uploading and editing of process models and conformance checking functionalities. We additionally aim to support event logs that are stored in a distributed environment.

REFERENCES

- [1] W. M. P. van der Aalst, *Process Mining - Data Science in Action, Second Edition*. Springer, 2016.
- [2] E. Verbeek, J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst, “ProM 6: The Process Mining Toolkit,” in *BPM Demonstration Track 2010, Hoboken, NJ, USA, September 14-16, 2010*, ser. CEUR Workshop Proceedings, M. L. Rosa, Ed., vol. 615. CEUR-WS.org, 2010.
- [3] M. L. Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, and L. García-Bañuelos, “APROMORE: An Advanced Process Model Repository,” *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7029–7040, 2011.
- [4] A. Berti, S. J. van Zelst, and W. M. P. van der Aalst, “Process Mining for Python (PM4Py): Bridging the Gap Between Process-and Data Science,” in *ICPM Demo Track 2019, Aachen, Germany, June 24-26, 2019.*, 2019, p. 13–16.
- [5] G. Janssenswillen, B. Depaire, M. Swennen, M. Jans, and K. Vanhoof, “bupaR: Enabling Reproducible Business Process Analysis,” *Knowl. Based Syst.*, vol. 163, pp. 927–930, 2019.
- [6] A. Berti and S. J. van Zelst, “PM4Py Web Services: Easy Development, Integration and Deployment of Process Mining Features in any Application Stack,” in *BPM Demonstration Track 2019, Vienna, Austria, September 1-6, 2019*, ser. CEUR Workshop Proceedings, vol. 2420. CEUR-WS.org, 2019, pp. 174–178.
- [7] M. Song and W. M. van der Aalst, “Supporting process mining by showing events at a glance,” in *Proceedings of the 17th Annual Workshop on Information Technologies and Systems (WITS)*, 2007, pp. 139–145.
- [8] V. Denisov, E. Belkina, D. Fahland, and W. M. P. van der Aalst, “The performance spectrum miner: Visual analytics for fine-grained performance analysis of processes,” in *BPM Demonstration Track 2018, Sydney, Australia, September 9-14, 2018*, ser. CEUR Workshop Proceedings, vol. 2196. CEUR-WS.org, 2018, pp. 96–100.
- [9] R. J. P. Mennens, R. Scheepens, and M. A. Westenberg, “A stable graph layout algorithm for processes,” *Comput. Graph. Forum*, vol. 38, no. 3, pp. 725–737, 2019.