bupaRflow: A Workflow Interface for bupaR

Brecht Steukers[1], Gert Janssenswillen[1,*], Gerhardus A. W. M. van Hulzen[1], Frank Vanhoenshoven[1] and Benoît Depaire[1]

[1]*UHasselt - Hasselt University, Faculty of Business Economics*
Agoralaan, 3590 Diepenbeek, Belgium

Abstract

In recent years, the open-source process analytics tool bupaR has seen a significant increase in usage. Among the advantages are its functional programming design – making it inherently suitable for interactive data analysis – and its reproducibility. However, writing scripts is still out of the comfort zone for many professionals who might benefit from the insights of process analysis. In order to make bupaR accessible to a wider audience, this paper presents bupaRflow, a graphical interface on top of bupaR that combines the workflow paradigm with an analytical building block architecture.

Keywords

process mining, event data, process analytics, functional programming, visual programming

1. Introduction

Since the publication of the first `R`-package for exploratory and descriptive analysis of event data in 2016 [1], the ecosystem of business process analytics in `R` has steadily grown in functionalities as well as user base [2, 3]. In general, the use of script-based tools for process analytics such as **bupaR** and **PM4Py** [4] has several advantages. Firstly, the product of the analysis is not just the results, but also the script that has led to these results, thereby making sure the analyses are perfectly reproducible. Secondly, scripts bring transparency to the table, as the steps undertaken in the analysis are explicitly made clear. Finally, it provides flexibility and extensibility, as the aforementioned tools are embedded within the data analytics ecosystems of `R` and `Python`.

These advantages, together with the fact that **bupaR** is available open-source, have contributed to its widespread use. Since the **bupaR** packages are freely available, they provide a perfect starting point for professionals to experiment with process mining and discover its value. However, the use of a programming language is still often regarded a considerable adoption barrier and can lead to steep learning curves. This makes script-based process analysis tools a viable option for professionals with programming experience, but less so for professionals without a programming background — or even a background in data analysis — who might also

BPM'22: Demo and Resources track, September 11–16, 2022, Münster, Germany

*Corresponding author.
brecht.steukers@student.uhasselt.be (B. Steukers); gert.janssenswillen@uhasselt.be (G. Janssenswillen); gerard.vanhulzen@uhasselt.be (G. A. W. M. van Hulzen); frank.vanhoenshoven@uhasselt.be (F. Vanhoenshoven); benoit.depaire@uhasselt.be (B. Depaire)
0000-0002-7474-2088 (G. Janssenswillen); 0000-0001-8962-9515 (G. A. W. M. van Hulzen); 0000-0003-2848-4492 (F. Vanhoenshoven); 0000-0003-4735-0609 (B. Depaire)

benefit from the insights delivered by process analysis.

In this paper, we present **bupaRflow** — a prototype graphical user interface built on top of **bupaR** to create process analysis workflows. By using the concept of functional building blocks — where each block represents a function, taking an input and turning it into an output — **bupaRflow** preserves the transparency provided using functional programming. The user of **bupaRflow** is able to perform process analysis using the core **bupaR** toolset, without the need for any programming. In addition, users session are saved so that analyses can be revisited and repeated at later moments.

Section 2 discusses the design principles and major features of **bupaRflow** . Section 3 describes its maturity, while Section 4 points to additional materials accompanying this demo, including a screencast, tutorial and instructions on how to access the tools. Section 5 concludes the paper and discusses avenues for future work.

2. Features

In the following paragraphs, we discuss the functionality (Sec 2.1), conceptual design (Sec 2.2) and architecture (Sec 2.3) of **bupaRflow** .

2.1. bupaR functionality

bupaRflow currently supports all functionalities provided by the core **bupaR** packages: bupaR (for main event log handling), processmapR (for creating directly-follow graphs and other visualizations), and edeaR (for calculating descriptive measures and event log filtering). Extensions towards other functionalities provided by the wider **bupaR**-ecosystem are planned to be added in the future. The architecture (see Section 2.3) is conceived in such a way that adding packages that comply with the design philosophy in *tidyverse* [5], can be integrated straightforwardly.

2.2. Conceptual Design

The starting point for the design of **bupaRflow** was to preserve the aforementioned unique qualities of script-based process analysis as much as possible, i.e. reproducibility, transparency, and flexibility. Coupled with the functional programming paradigm that is used by **bupaR** it followed naturally to take a visual programming approach, where each function forms an analytical building block. When connected, these blocks form analytical workflows.

The set of workflows illustrated in Figure 1 perform several analysis on the example *patients* dataset. Two different process maps are made, with different configurations (cannot be observed in the screenshot). Furthermore, the data is filtered on trace frequency, after which throughput times are calculated and plotted. Furthermore, the filtered traces are shown using the trace explorer. For more information on these workflows, we refer to the tutorial and screencast.

In the field of data science, this visual programming approach is mostly known from tools as KNIME [6] and RapidMiner [7]. It should be noted that an extension to RapidMiner for process mining, called RapidProM [8], exists. However, as the main focus of **bupaRflow** is to make process analysis more accessible to professionals without a programming or even data analysis background, it was specifically decided to create a standalone, dedicated application rather

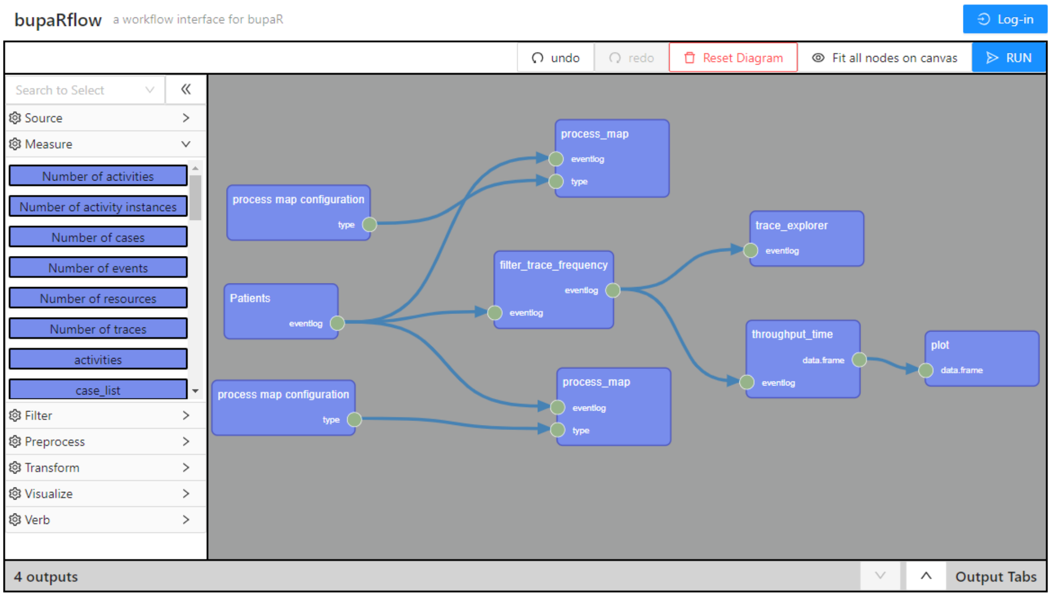

Figure 1: Screenshot of **bupaRflow** with a set of workflows.

than an extension to one of the existing tools, as the latter might by unfamiliar and thus form another barrier to be overcome.

Using this visual programming approach preservers the transparency that comes with script-based process analysis. User management allows the analysis to be saved and revisited later. However, some flexibility and extensibility is sacrificed. Adding new blocks by users is not possible, while combining **bupaR-** functionalities with other libraries is only possible if these are explicitly included in the applications. Currently, this is only done for functionalities of the *tidyverse* [5], the usage of which can be seamlessly integrated with **bupaR**. In contrast, an advantage that **bupaRflow** has over **bupaR** itself is that it allows parts of analysis workflows to be reused by creating of several branches after a specific block, as can be seen in Fig. 1.

It should be noted that this approach is different from PMTK, the web-based process mining tool on top of **PM4Py** , which does not use a visual programming approach but provides an analysis toolkit using a dashboard approach, not unlike existing commercial tools.

2.3. Architecture

bupaRflow is conceived as a web application using an API to **bupaR** in the back-end. A conceptual overview of the architecture can be seen in Figure 2. The interactive web interface was created using **Vue.js** [9] while the API was created using **plumber** [10]. In order to enhance performance, the app allows users to indicate whether a specific block should be treated as persistent, i.e. so that it will not be recomputed at each run. Firebase is used to store the data. It should be noted that the back-end is made in such a way that new functions can be added with minimal effort — i.e. by adding them to a configuration. The app is currently hosted on Azure.

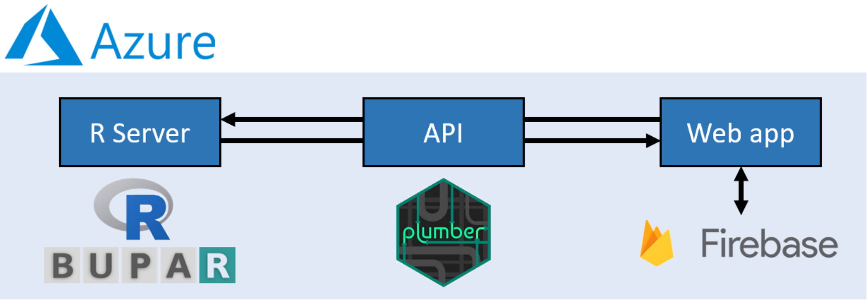

Figure 2: Overview of **bupaRflow** architecture.

3. Maturity

The **bupaRflow** tool presented in this paper should be regarded as a first prototype. It has not been made publicly available before, and as such case studies using the tool are not available yet. Nonetheless, it stands upon the foundation of the **bupaR**-ecosystem. Since its conception, the **bupaR**-ecosystem has amassed more than 800K downloads in 158 countries across the globe, thereby encouraging the further adoption of process mining. The user base of **bupaR** is highly varied, ranging from both service and product industries, governmental agencies, as well as NGOs. Over the years, a considerable amount of research papers and case studies using **bupaR** have been published. [11, 12, 13, 14]

4. Further materials

For reviewing purposes, **bupaRflow** has been made available via this link: https://buparflow.azurewebsites.net/. It can be tested anonymously by using the *Proceed without an account* option. A 4-minute screencast is available here: https://tinyurl.com/bpmbuparflow. A tutorial can be found here: https://gertjanssenswillen.github.io/bpmbuparflowdemo

5. Conclusions and Future Work

This paper presented **bupaRflow**, a web application that allows the use of **bupaR-** functionalities using visual programming. It is targeted to professionals without any background in data analysis or programming, who want to discover how process mining can bring additional insights to their conventional analyses.

The tool as presented in this paper is a prototype, and several improvements are foreseen for the future. While user management is in place, it currently only allows saving a single canvas. In order to improve the user experience, the design of the interface needs further improvement, and proper error handling needs to be provided. Next to the further addition of functionalities beyond the **bupaR-** core, also functionalities to export data and save outputs need to be provided. Additional functionalities outside of the **bupaR-** ecosystem, for instance for data import, can be considered as well.

References

[1] M. Swennen, G. Janssenswillen, M. Jans, B. Depaire, K. Vanhoof, Capturing process behavior with log-based process metrics, in: Proceedings of the 5th International Symposium on Data-driven Process Discovery and Analysis, CEUR Workshop Proceedings, RWTH Aachen University, 2015, pp. 141–144.

[2] G. Janssenswillen, F. Mannhardt, M. Creemers, B. Depaire, M. Jans, L. Jooken, N. Martin, G. Van Houdt, Extensions to the bupaR ecosystem: An overview, in: Proceedings of the ICPM Doctoral Consortium and Tool Demonstration Track, CEUR Workshop Proceedings, 2020, pp. 43–46.

[3] G. Janssenswillen, B. Depaire, M. Swennen, M. Jans, K. Vanhoof, bupaR: Enabling reproducible business process analysis, Knowledge-Based Systems 163 (2019) 927–930.

[4] A. Berti, S. J. Van Zelst, W. van der Aalst, Process mining for python (pm4py): bridging the gap between process-and data science, arXiv preprint arXiv:1905.06169 (2019).

[5] H. Wickham, The tidyverse, R package ver 1 (2017) 1.

[6] M. R. Berthold, N. Cebron, F. Dill, T. R. Gabriel, T. Kötter, T. Meinl, P. Ohl, K. Thiel, B. Wiswedel, Knime-the konstanz information miner: version 2.0 and beyond, AcM SIGKDD explorations Newsletter 11 (2009) 26–31.

[7] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, T. Euler, Yale: Rapid prototyping for complex data mining tasks, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006, pp. 935–940.

[8] W. M. van der Aalst, A. Bolt, S. J. van Zelst, Rapidprom: mine your processes and not just your data, arXiv preprint arXiv:1703.03740 (2017).

[9] E. You, Vuejs framework, 2020.

[10] B. Schloerke, J. Allen, plumber: An API Generator for R, 2022. Https://www.rplumber.io, https://github.com/rstudio/plumber.

[11] N. A. Uzir, D. Gašević, J. Jovanović, W. Matcha, L.-A. Lim, A. Fudge, Analytics of time management and learning strategies for effective online learning in blended environments, in: Proceedings of the tenth international conference on learning analytics & knowledge, 2020, pp. 392–401.

[12] K. K. Larsson, Digitization or equality: When government automation covers some, but not all citizens, Government Information Quarterly 38 (2021) 101547.

[13] H. Nguyen, K. Y. Lim, L. L. Wu, C. Fischer, M. Warschauer, "we're looking good": Social exchange and regulation temporality in collaborative design, Learning and Instruction 74 (2021) 101443.

[14] J. González-García, C. Tellería-Orriols, F. Estupiñán-Romero, E. Bernal-Delgado, Construction of empirical care pathways process models from multiple real-world datasets, IEEE journal of biomedical and health informatics 24 (2020) 2671–2680.