VisuELs: Visualization of Event Logs (Extended Abstract)

Gaël Bernard
University of Toronto
Faculty of Information
Toronto, Canada
gael.bernard@utoronto.ca

Periklis Andritsos
University of Toronto
Faculty of Information
Toronto, Canada
periklis.andritsos@utoronto.ca

***Abstract*—We propose a technique to transform event logs of any size into compact visualizations that we call VisuELs (Visualization of Event Logs). VisuELs are particularly useful in the exploratory phase of a process mining project to extract key insights about an event log (e.g., average length, top activities, patterns of behaviours). New VisuELs can be generated using Python or a web-based tool without fine-tuning any parameter.**

***Index Terms*—process mining, sampling, visualization**

I. Introduction

Process mining aims to extracts insights from event logs. Due to the complex nature of many event logs, process mining is often conducted in an explorative way [1]. Recently, Zerbato et al. investigated why and how process analysts explore logs in practice [2]. In a nutshell, they aim to *"get a feeling of how complex the data is"* and *"become familiar with the data and the process before determining any direction"*. This task, common to many data science projects, is often referred to as *profiling*. Typically, practitioners perform event logs profiling by alternating between various views offered by academic and commercial tools, e.g., histograms, Fuzzy Miner [3], dotted chart [4], or chords diagram [5]. We aim to propose a new view to assist process analysts in the profiling of event logs.

II. VisuELs

VisuELs aims to be compact and readable–no matter the input logs' complexity, size, or traces' length. To achieve this, we show only a few representative traces. Similar to the dotted chart, the rows show the representatives, and the (coloured) squares represent the (type of) activities. In Fig. 1, we show a VisuEL of the Sepsis event logs composed of 1 049 cases. Despite the relative simplicity of the representation, a VisuEL achieves four ambitious goals. VisuELs should: (G1) summarize the logs; (G2) be easy to interpret; (G3) be easy to build; (G4) be comparable. Next, we present five features that contribute to the fulfillment of these goals.

Downsizing Scale. To choose the number of representatives to display on VisuELs, we propose the following downsizing scale: $\lceil log_{1.5}(s) \rceil$, s being the number of cases of the original event logs. Typically, a log of 10 000 traces is summarized by 23 representatives. Even extremely large or small logs would fit in a grid of reasonable size; e.g., $\lceil log_{1.5}(10^9) \rceil = 52$ and $\lceil log_{1.5}(3) \rceil = 3$. Similar to reducing VisuELs' vertical extent,

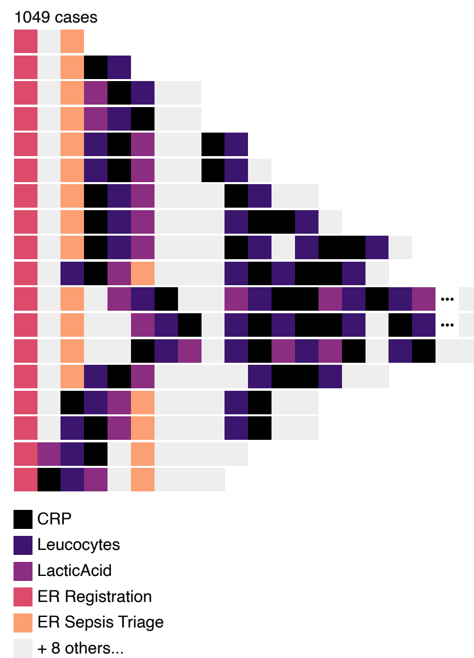

Fig. 1. VisuEL summaring the Sepsis event logs. These 18 traces are the best representative of the original 1 049 traces.

we limit the horizontal size by showing a maximum of 20 activities. As can be seen in Fig. 1, suspension points highlight the presence of longer traces. These methods to limit VisuELs' sizes enable the visualization of logs of any size in a readable way (G1) and make their comparison possible (G4).

Sampling. To select the traces that will appear on the VisuEL, we borrow the *iterative c-min sampling* technique that has been shown to produce the most representative downsized event logs–in terms of earth movers' distance from the original event logs [6]. Using this technique, we ensure that a VisuEL fairly represents the input logs (G1).

Colors and Legend. For readability purposes (G2), we colour only the top 5 activities and use a neutral gray colour for the other activities. In addition, we added an option to produce several VisuELs using a single *shared legend*. This way, distinct VisuELs will use the same colours for the same activities, making their comparison easier (G4). The advantage of the shared legend is highlighted in the second use case, where we visualize clusters of traces.

Ordering. The traces are sorted by similarity to facilitate their reading (G2). To achieve this, we measure their Levenshtein distance, and then we apply an approximation of the travelling salesman problem to find the ordering that minimizes the distance. Ultimately, similar traces will appear next to each other, which facilitates the identification of patterns.

Parameter Free. To make the creation of VisuELs flawless (G3), we ensure that it is possible to create VisuELs without having to fine-tune any parameters.

III. Use Case

We showcase VisuELs using two use cases.

A. Logs Gallery

We transformed 18 mainstream datasets from process mining into VisuELs. Due to space constraints, only one of them is visible in Fig.1, while the other ones are visible online[1]. The 18 VisuELs provide a clear overview of the datasets from where we can extract insights such as the occurrence of loops of size 1 (BPI 2017), traces often starting with the same set of activities (BPI 2012), a broad set of unique activities (BPI2018), few variants appearing many times (BPI2020_1), or short traces (BPI2020_5).

B. Clusters of traces

We used ngrams and KMeans to discover 12 clusters of similar traces from the dataset BPI 2020 competition (Permit Log). The goal was to highlight the ability of VisuELs to summarize the clustering results. In Fig. 2, we show the original Permit Log and 4 clusters–all the clusters are visible online[1]. We used a *shared legend* to ease their comparison. Overall, we can extract valuable insights from the VisuELs visible in Fig. 2. First, clusters 4 and 5 seem to be relatively structured. Second, cluster 5 does not have the activity 'declaration submitted by employee' compared to other clusters. Third, cluster 8 look very chaotic and lengthy. Fourth, in cluster 9, the activity 'declaration submitted by employee' occurs 3 times per trace, a behaviour specific to this cluster. We argue that such observations may be difficult to extract if one has to switch between various views for each cluster.

IV. Architecture and Scalability

VisuEL is written in Python and produces scalable vector graphics (SVG). It can read several formats including XES files [7] and PM4py object [8]. Moreover, we also propose a web-based tool version. The source code, the web tool, and an introductory video are available online[1].

VisuELs are fast to generate, even for large event logs. The longest time to build a VisuEL in our use case was for the 'BPI 2018' dataset composed of 2.5M events, where it took 42 seconds using a machine with 16GB of RAM, 4 CPUs, and a processor speed of 2.8 GHz. The time can be further reduced using extra heuristics, but such optimization is not within the scope of this paper.

Fig. 2. VisuELs of the BPI 2020 (Permit Log). The top left shows the dataset 'Original Dataset', and 4 clusters (out of 12) are shown.

[1]https://visuel.customer-journey.me

References

[1] M. F. Sani, S. J. van Zelst, and W. M. van der Aalst, "The impact of biased sampling of event logs on the performance of process discovery," *Computing*, pp. 1–20, 2021.

[2] F. Zerbato, P. Soffer, and B. Weber, "Initial insights into exploratory process mining practices," in *Business Process Management Forum*, A. Polyvyanyy, M. T. Wynn, A. Van Looy, and M. Reichert, Eds. Cham: Springer International Publishing, 2021, pp. 145–161.

[3] C. W. Günther and W. M. Van Der Aalst, "Fuzzy mining–adaptive process simplification based on multi-perspective metrics," in *International conference on business process management*. Springer, 2007.

[4] 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.

[5] A. Jalali, "Reflections on the use of chord diagrams in social network visualization in process mining," in *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2016, pp. 1–6.

[6] G. Bernard and P. Andritsos, "Selecting representative sample traces from large event logs," in *International Conference on Process Mining*. Springer, 2021.

[7] "IEEE standard for extensible event stream (xes) for achieving interoperability in event logs and event streams," *IEEE Std 1849-2016*, 2016.

[8] 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," *CoRR*, vol. abs/1905.06169, 2019.