VisuEL: Visualization of Event Logs

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) without fine-tuning any parameter. New VisuELs can be generated flawlessly through Python or using a web-based tool.**

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

I. Introduction

The discipline of 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]. In a recent study, [2], Zerbato et al. investigated why and how process analysts explore logs in practice. In a nutshell, process analysts 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 any data science projects, is often referred to in the literature as *profiling*. In process mining, practitioners execute this task by alternating between various views offered by the academic and commercial tools; e.g., Fuzzy Miner, histograms, dotted chart. We aim to propose a new view to to assist process analysts during the profiling phase of a process mining project.

II. VisuELs

VisuELs stands for Visualization of Event Logs. This new way to visualize an event log aims to be compact and readable–no matter the input logs' complexity, size, or traces' length. In addition, this technique is parameter-free to make the creation of new VisuELs effortless. To achieve this, we select a few representative traces to summarize the original event logs. Similar to the dotted chart, the rows show the representatives, and the (colored) squares represent the (type of) activities. In Fig. 1, we show a VisuEL of the Sepsis event logs, a log originally composed of 1 049 cases with an average length of 14.5. Despite the relative simplicity of the representation, a VisuEL achieves four ambitious goals. VisuELs should: (G1) clearly summarize the logs; (G2) be easy to interpret; (G3) be easy to build; and (G4) be comparable. Next, we present five features that contribute to the fulfillment these goals.

Downsizing Scale. To choose the number of representatives to display on VisuELs, n, we propose the following downsizing scale: $n = \lceil log_{1.5}(s) \rceil$, s being the size of the original

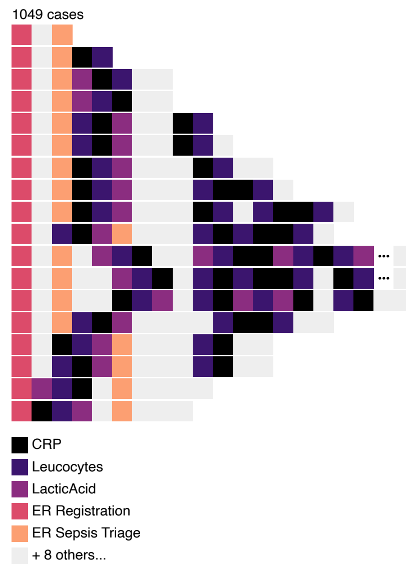

Fig. 1. VisuEL summaring the Sepsis event logs. These 17 traces are the best representative of the original 1049 traces.

event logs (i.e., number of cases). Typically, using this scale, a log of 5000 traces is summarized by 21 representatives. Even extremely large or small logs would fit in a grid of reasonable size; e.g., $\lceil log_{1.5}(10^9) \rceil = 51$ and $\lceil log_{1.5}(4) \rceil = 3$. Similar to how we reduce VisuELs' vertical extent, 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 allow us to depict event logs of any size in a readable way (G1) and make their comparison possible (G3).

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 [3]. 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. This reduces the size of the legend visible under the representative traces. 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 (G3). We emphasized the advantage of the shared legend in the second case study, showing how to visualize clusters of similar 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 and make the identification of patterns easier.

Parameter Free. To make the creation of VisuELs flawless (G2), we ensure that it is possible to create VisuELs without having to fine-tune any parameters.

III. Use Case

We show the value of VisuELs by using them to depict 18 datasets in a logs gallery and 12 clusters of similar traces.

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. Clustering

We extracted 12 clusters of traces from the dataset BPI 2020 competition (Permit Log) using ngrams and KMeans. The clustering is not part of VisuELs since we want VisuELs to be independent of any preprocessing. Then, we leverage VisuELs to inspect the clusters. In Fig. 2, we show the original Permit Log and 4 of the 12 clusters–all the clusters are visible online[1]. Note that we used the feature allowing several VisuELs to share a single legend to ease their comparison. 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 [4] and PM4py object [5]. Moreover, we propose a web-based tool that flawlessly transforms a CSV or a XES file into a VisuEL. Note that we parse the data on the customer side (using Javascript) to send a limited amount of data to the server. The source code and the web service, as well as an introductory video are available online[1].

VisuELs are relatively fast to generate, even for large event logs. The longest time to build one of the 18 logs from the case study 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. This time can be reduced using heuristics, but such optimization is not within the scope of this paper.

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

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.

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

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

[5] 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.