



# Visualizing Business Process Evolution

Anton Yeshchenko<sup>(✉)</sup> , Dina Bayomie , Steven Gross, and Jan Mendling 

Vienna University of Economics and Business, Vienna, Austria  
{anton.yeshchenko, dina.sayed.bayomie.sobh, steven.gross,  
jan.mendling}@wu.ac.at

**Abstract.** Literature in business process research has recognized that process execution adjusts dynamically to the environment, both intentionally and unintentionally. This dynamic change of frequently followed actions is called process drift. Existing process drift approaches focus to a great extent on drift point detection, i.e., on points in time when a process execution changes significantly. What is largely neglected by process drift approaches is the identification of temporal dynamics of different clusters of process execution, how they interrelate, and how they change in dominance over time. In this paper, we introduce process evolution analysis (PEA) as a technique that aims to support the exploration of process cluster interrelations over time. This approach builds on and synthesizes existing approaches from the process drift, trace clustering, and process visualization literature. Based on the process evolution analysis, we visualize the interrelation of trace clusters over time for descriptive and prescriptive purposes.

**Keywords:** Process drift · Process change · Process evolution · Process trace clustering

## 1 Introduction

Business process management (BPM) has long researched business processes through a prescriptive lens. This perspective emphasizes the static nature of processes, their design, and their execution. The descriptive perspective, on the other hand, acknowledges the dynamic nature of processes and their intentional and unintentional adjustment over time due to changes in the environment, technological capabilities, seasonal differences, and other factors [1, 2]. Similarly, literature in routine research has increasingly recognized the dynamic change of routine executions over time [3, 4].

To address this dynamism, recent research aims to identify changes in processes over time. Most prominently, process drift uses event logs to detect points in time when changes take place [1, 5]. While process drift offers insights into when changes in the process happen, it does not provide information on how different executions of the same process interrelate with each other, in particular, when and how similar process executions occur and dominate. Thus, the meaningful interpretation of process evolution over time, i.e., the changes in the process execution over time, remains a research challenge. Evolution is defined as a recurring variation, selection, and retention among entities of a designated population [6]. We build on this concept by declaring all traces

generated during the execution of a process as the designated population, while the variation, selection, and retention (i.e. the appearance, disappearance, and change) happens among sub-groups of these traces. This is due to scarce environmental resources [6] which correspond to the organizational resources available for the execution of the process.

In order to advance our understanding of business process evolution, this research aims to answer the following research question: *How can temporal dynamics of different process executions be analyzed?* To approach this question, we propose a process evolution analysis (PEA) method that uses process execution clustering and time-series line plots to study the evolution of the process execution clusters over time. PEA thereby aims to show insights on how clusters of process executions are interrelated overtime on a process evolution graph.

This paper proceeds as follows. Section two discusses the main concepts. Section three describes our research method. Section four presents the results of applying PEA. Section five concludes with a discussion of the results, limitations of the approach, and future research directions.

## 2 Background

In this section, we describe the main concepts used in this paper. First, we describe and emphasize the difference in our approach to process drift. Next, we outline the process mining method to find trace clustering. It is used to reduce the complexity of process mining problems by grouping the process instances based on their similarity, i.e., trace fitness, execution time, etc. Lastly, we summarize the state of the art of different visualization techniques that influenced our technique.

### 2.1 Process Drift

Process drift refers to a change in processes executions and is known as a key challenge in process mining [7]. Process drift is extensively studied in the process mining literature [1, 5, 8, 9] with a focus on drift points, drift types [1, 8], and drift visualizations [5, 9]. Seeliger et al. [10] define process drift as a significant behavioral change during the execution of a process that has occurred over time. A drift in a process, according to their definition, occurs when almost all traces are influenced by that drift [10]. Similarly, Maaradji et al. [1] focus on the detection of process drifts by identifying statistically significant differences between the observed process behavior of different intervals in time. Yeshchenko et al. [5] focus on the detection of separate behavioral drifts within a process that are simultaneously present in the event log.

Most existing process drift research interprets process change as a change of the underlying control flow (implicit process model change). However, changes in the process can also be viewed through various process performance measures that are important for the analysis of the process executions [11]. These measures show how the process characteristics (such as cycle time, cost per instance, etc.) change over time. Existing process drift approaches thereby focus on the discovery of behavioral change within the event log, ignoring the impact on process measures. That creates a process overview

where the analyst might be incapable of inferring what change in the process is important enough to manage. Our method allows for tracking the process KPIs with correspondence to the change.

## 2.2 Trace Clustering

Trace clustering techniques aim to reduce the complexity of process mining problems [12–15]. These techniques simplify an event log by creating groups of traces based on some trace similarity metric. The clustering procedure ensures that the clusters have minimal in-cluster distance between traces based on that metric. The metrics used in trace clustering range from the structural similarity between traces (i.e., based on fitness [12]) and performance-related, such as the cost per instance and the cycle time [15].

One use case of trace clustering algorithms is to simplify the problem of process discovery on sparse event logs to provide more accurate and comprehensive results [12]. Process mining literature emphasizes the use of trace clustering for the discovery of models which are based on the clusters. Instead, our approach directly explores the interrelations of different clusters using visualization techniques.

## 2.3 Visualization of Change

Process mining research focuses on discovering precise process models rather than exploring the change in processes. On the other hand, the visualization field is rich with approaches to analyze such data [16]. The canonical example of visualizing change is line plotting, where the x-axis corresponds to a time, and the y-axis displays the selected measure. More recently, developments on these types of graphs are visualization of opinion change in social media [17–22]. These approaches are built upon the plots that emphasize the change of variables with time and include line plots [18, 20, 21], theme river [19, 21], and Sankey diagram [17–20] representations. Our approach is influenced by the ideas from the field to represent the change of several time-series.

## 3 Process Evolution Analysis Technique

Figure 1 shows an overview of our proposed process evolution analysis (PEA) technique, which visualizes the change of the process execution behavior over time. PEA takes an event log as input and produces a time series line chart as output. PEA has three steps to generate the required visualization. The first step is using trace clustering to partitioning the log, then compute some process metric over the clusters (e.g., the number of cases for each cluster). Finally, PEA visualizes the clusters and their process metric values to show the process evolved.

*Trace clustering* – an event log is clustered into several clusters in order to find distinct regions in the process behavior. There are different techniques used to cluster an event log [14]. A simple cluster technique is partitioning the log based on time window interval. For instance, cases that occurred in the first quarter of the year are in the same cluster. Although, cases that occurred in the last quarter are in another cluster. One established clustering technique is based on trace similarity to achieve high fitness

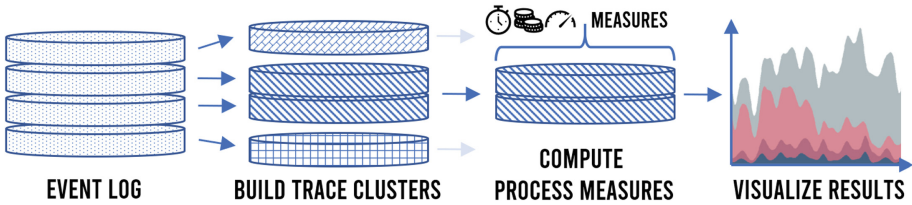


Fig. 1. Overview of the proposed approach

between the cases within the same cluster [12, 13]. Another technique to cluster the log is using the cycle time to distribute the cases over the clusters [15]. Thus, all the cases that have a long cycle time are in the same cluster, while cases with a short cycle time are in another cluster.

*Process evolution graphs* – We use the time-series line plot [18, 20, 21] and theme river [19, 22] visualization in order to analyze the change in the behavior between the different clusters. The graph shows the change of a process metric over time for each trace cluster. Different process execution characteristics can be measured; for instance, the number of executed cases per day, the average cycle time of cases per month, or the average resource workload per day.

By combining trace clustering with the computation of process metrics and visualizing the result on a time-series plot, behavioral changes in the process execution can be made visible.

## 4 Findings

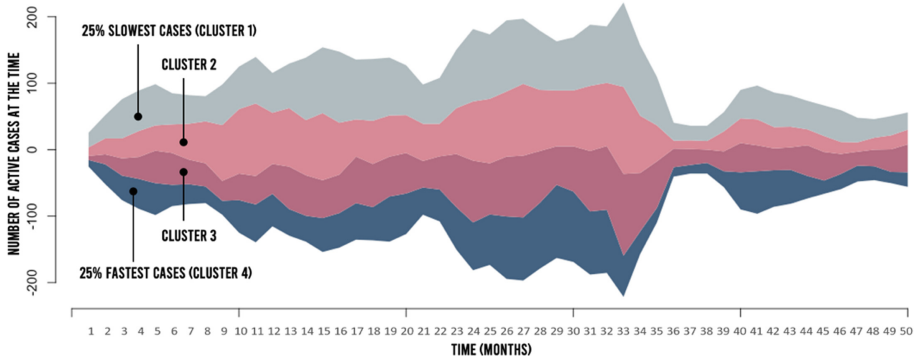
*Experiment setup* – We conduct two experiments on two datasets and visualize process evolution using different trace clustering techniques. The first experiment uses the log from the help desk of an Italian software company<sup>1</sup>. The traces are clustered based on the cycle time per trace with the python script, made available on GitHub<sup>2</sup>. The clusters are used to visualize process evolution using a theme-rivers chart. Each generated cluster represents a different cycle time behavior. The second experiment uses the traffic logs from BPIC<sup>3</sup>. We applied a trace clustering based on trace fitness using ActiTrace ProM plugin [12] with the following settings: target ICS fitness 0.9, the maximal number of clusters is 6, and add the remaining traces to other clusters. Each generated cluster represents a different process behavior. Moreover, we visualize process evolution using a time-series line chart.

For the first experiment, Fig. 2 shows the number of cases for each cluster over 50 months. The x-axis depicts the months within this timespan, while the y-axis represents the number of active cases. As has been described before, the traces were clustered according to the cycle time of each case. The 25% of cases with the slowest cycle time

<sup>1</sup> <https://data.4tu.nl/repository/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb>.

<sup>2</sup> <https://github.com/yesanton/Visualizing-Business-Process-Evolution>.

<sup>3</sup> <https://data.4tu.nl/repository/uuid:270fd440-1057-4fb9-89a9-b699b47990f5>.



**Fig. 2.** Number of cases for each cluster over time for *help desk* event log

are in Cluster 1. Cluster 2 and 3 follow to cluster 1 in steps of 25%. Cluster 4 contains 25% of cases, which were the fastest.

The graph can be analyzed in several ways. First, the *cluster containing the most active cases at a certain point in time* can be identified. For instance, cluster 1 (25% slowest cases in the process) contains the most active cases in the first five months. In the sixth month, however, cluster 2 is dominating the other clusters, until most cases can be found in cluster 1 from month 15 to 26. Second, the *discrepancy in case distributions* between the different clusters can be observed. One example is the timespan from the 7<sup>th</sup> to 20<sup>th</sup> month, in which there is a substantial discrepancy of active cases for the clusters (i.e., the number of cases in each cluster differ considerably) compared to the timespan between the 24<sup>th</sup> and 28<sup>th</sup> month (i.e., the clusters contain a similar amount of cases). Third, it can be seen whether the number of cases for the *clusters correlates over time*. For instance, all clusters in the timespan between the 33<sup>rd</sup> and 36<sup>th</sup> month decrease similarly in number. However, between the 27<sup>th</sup> and 32<sup>nd</sup> month, cluster 3 and cluster 4 negatively correlate. Fourth, *quick changes in a cluster* can be spotted. For instance, cluster 2 is more than doubled its active cases within a quarter of a year from month 7 onwards.

For the second experiment, Fig. 3 shows the number of cases for each cluster over 56 months. Differently to the previous experiment, the clusters were not generated by grouping traces with similar cycle time but based on their trace fitness (ICS-fitness). Six clusters were generated. It can be seen that cluster 1 dominated all other clusters in terms of active cases until month 41. From this point in time, the cluster does not contain enough active cases to be visible on the graphical representation. Until the 41<sup>st</sup> month, cluster 1 also seems to correlate with cluster 2, while the magnitude of changes for cluster 1 is much bigger. Sudden changes are also visible. In month and 28, cluster 1 increases substantially in the number of active cases within one month by the factor 3 to 4. Clusters 3, 4, 5, and 6 are having fewer cases than 250 for most of the time; thus, they are not paying a significant role during the execution of the process.

Cluster 3 is dominating clusters 4, 5, and 6 in the period between the 1<sup>st</sup> to 7<sup>th</sup> and 53<sup>rd</sup> to 55<sup>th</sup>. Aside from that period, no clear dominance can be found within these four clusters. Cluster 5 seems to be inactive for most of the time.

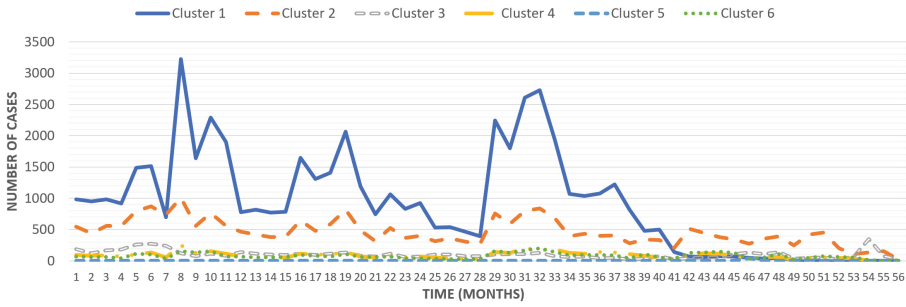


Fig. 3. Number of cases for each cluster overtime for *traffic fines* event log

## 5 Discussion and Conclusion

In this paper, we described the process evolution analysis (PEA) method, which visualizes different clusters of process executions and their interrelation over time. PEA builds on the concepts of trace clustering, process drift, and process visualization.

PEA has several practical and theoretical implications. From a practical viewpoint, PEA can be used as a technique to analyze process executions and to identify undesired behaviors that emerge. For instance, if the clusters are generated by categorizing cases based on their cycling time, a rise of cases with a long cycling time can be identified, and take the required actions. Thereby, PEA can be used for real-time analysis of process execution to intervene directly, or as a retrospect, analysis to identify recurring patterns of undesired process behaviors. The same applies to clusters, which were generated by their trace fitness. The underlying process models from these clusters can be generated through process discovery (e.g., by the use of a process mining tool) and thereby unwanted process executions discovered. The evolution of these clusters can again be made explicit through PEA.

From a theoretical point, this is a first step towards synthesizing research on trace clustering, process drift, and process visualization into one technique. By doing so, we enabled the identification and analysis of changes within the different clusters of traces over time. While process drift focuses mainly on points in time in which the general process behavior changes [10], PEA shows how these underlying behavioral changes interrelate over time. It also extends existing research on process clustering, which has mainly focused on discovering the underlying process models of these clusters for further analysis.

We recommend conducting future research in the following directions. First, alternative forms of visualization could be investigated. One such promising visualization is the stacked graphs, which is designed to facilitate the identification of patterns, trends, and unexpected occurrences [23]. Future research should investigate which forms of visualization for PEA are useful, taking into consideration different application scenarios. Second, different information besides the number of active cases within a cluster could be plotted on the graph. For instance, the y-axis could depict the average cycle time for active cases, which start at a specific time. Third, other forms of clustering the traces could be explored. In this paper, we clustered the traces according to their trace fitness

and the cycle time of the cases. One alternative could be to cluster the traces according to the costs or revenue of the cases to investigate the evolution of these clusters.

## References

1. Maaradji, A., Dumas, M., La Rosa, M., Ostovar, A.: Fast and Accurate Business Process Drift Detection. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-23063-4>
2. Pentland, B., Recker, J., Kim, I.: Capturing reality in flight ? Empirical tools for strong process theory. In: Thirty Eighth International Conference on Information Systems, pp. 1–12, Seoul (2017)
3. Feldman, M.S., Pentland, B.T.: Reconceptualizing organizational routines as a source of flexibility and change. *Adm. Sci. Q.* **48**, 94 (2003). <https://doi.org/10.2307/3556620>
4. Pentland, B.T., Feldman, M.S., Becker, M.C., Liu, P.: Dynamics of organizational routines: a generative model. *J. Manag. Stud.* **49**, 1484–1508 (2012). <https://doi.org/10.1111/j.1467-6486.2012.01064.x>
5. Yeshchenko, A., Di Ciccio, C., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: Laender, A.H.F., Pernici, B., Lim, E.-P., de Oliveira, J.P.M. (eds.) ER 2019. LNCS, vol. 11788, pp. 119–135. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-33223-5\\_11](https://doi.org/10.1007/978-3-030-33223-5_11)
6. van de Ven, A.H., Poole, M.S.: Explaining development and change in organizations. *Acad. Manag. Rev.* **20**, 510–540 (1995). <https://doi.org/10.2307/258786>
7. van der Aalst, W., et al.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM 2011. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-28108-2\\_19](https://doi.org/10.1007/978-3-642-28108-2_19)
8. Maaradji, A., Dumas, M., Rosa, M.La, Ostovar, A.: Detecting sudden and gradual drifts in business processes from execution traces. *IEEE Trans. Knowl. Data Eng.* **29**, 2140–2154 (2017). <https://doi.org/10.1109/TKDE.2017.2720601>
9. Denisov, V., Belkina, E., Fahland, D., Van Der Aalst, W.M.P.: The performance spectrum miner: visual analytics for fine-grained performance analysis of processes. In: CEUR Workshop Proceedings, vol. 2196, pp. 96–100 (2018)
10. Seeliger, A., Nolle, T., Mühlhäuser, M.: Detecting concept drift in processes using graph metrics on process graphs. In: ACM International Conference Proceeding Series Part F1271 (2017). <https://doi.org/10.1145/3040565.3040566>
11. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Heidelberg (2018). <https://doi.org/10.1007/978-3-642-33143-5>
12. De Weerd, J., Vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. *IEEE Trans. Knowl. Data Eng.* **25**, 2708–2720 (2013). <https://doi.org/10.1109/TKDE.2013.64>
13. De Leoni, M., Van Der Aalst, W.M.P., Dees, M.: A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs. *Inf. Syst.* **56**, 235–257 (2016). <https://doi.org/10.1016/j.is.2015.07.003>
14. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace clustering in process mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) BPM 2008. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-00328-8\\_11](https://doi.org/10.1007/978-3-642-00328-8_11)
15. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace clustering based on conserved patterns: towards achieving better process models. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 170–181. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-12186-9\\_16](https://doi.org/10.1007/978-3-642-12186-9_16)

16. Aigner, W., Miksch, S., Schumann, H., Tominski, C.: Visualization of Time-Oriented Data. Springer, London (2011). <https://doi.org/10.1007/978-0-85729-079-3>
17. Liu, S., Wu, Y., Wei, E., Liu, M., Liu, Y.: StoryFlow: tracking the evolution of stories. IEEE Trans. Vis. Comput. Graph. **19**, 2436–2445 (2013). <https://doi.org/10.1109/TVCG.2013.196>
18. Cui, W., Liu, S., Wu, Z., Wei, H.: How hierarchical topics evolve in large text corpora. IEEE Trans. Vis. Comput. Graph. **20**, 2281–2290 (2014). <https://doi.org/10.1109/TVCG.2014.2346433>
19. Sung, C.Y., Huang, X.Y., Shen, Y., Cherng, F.Y., Lin, W.C., Wang, H.C.: Exploring online learners' interactive dynamics by visually analyzing their time-anchored comments. Comput. Graph. Forum. **36**, 145–155 (2017). <https://doi.org/10.1111/cgf.13280>
20. Liu, S., Yin, J., Wang, X., Cui, W., Cao, K., Pei, J.: Online visual analytics of text streams. IEEE Trans. Vis. Comput. Graph. **22**, 2451–2466 (2016). <https://doi.org/10.1109/TVCG.2015.2509990>
21. Wu, Y., Liu, S., Yan, K., Liu, M., Wu, F.: OpinionFlow: visual analysis of opinion diffusion on social media. IEEE Trans. Vis. Comput. Graph. **20**, 1763–1772 (2014). <https://doi.org/10.1109/TVCG.2014.2346920>
22. Havre, S., Hetzler, B., Nowell, L.: ThemeRiver: visualizing theme changes over time. In: Proceedings of IEEE Symposium on Information Visualization, pp. 115–123 (2000). <https://doi.org/10.1109/infvis.2000.885098>
23. Byron, L., Wattenberg, M.: Stacked graphs – geometry & aesthetics. IEEE Trans. Vis. Comput. Graph. **14**, 1245–1252 (2008). <https://doi.org/10.1109/TVCG.2008.166>