Clustering Event Logs based on Behavioral Similarity

Agnes Koschmider[1]

Abstract: This paper summarizes an approach for clustering of event logs based on the notion of behavioral similarity. Additionally, behavioral similarities between events are considered instead of a pure sequence of events allowing to identify homogeneous traces. Different to existing approaches this technique constructs a classification a-priori allowing to directly detecting changes in historical and real-time traces. The evaluation of our approach demonstrates efficiency in terms of change detection, performance time and the quality of clusters.

Keywords: Event logs, process mining, clustering, flexible processes

1 Motivation & Approach

The analysis of human behavior in the context of intelligent, connected environments is a challenging task. Humans behave according to best practices and a single behavioral model is typically not sufficient to represent them all. In fact, existing process mining algorithms reportedly generate spaghetti models from event logs of flexible processes, which are largely incomprehensible. One solution for event log structuring and analysis in such environments provides clustering techniques, which intend to split log files according to a notion of similarity or dissimilarity so that the same or similar tasks or processes are contained in each group and thus smaller and simpler process models can be mined. This paper summarizes a novel approach for clustering event logs by their behavioral similarity, rather than deriving a unique process model encompassing all traces. The clustering approach takes a process model as input but then iteratively constructs a-priori a classification and assigns traces to groups that are found most similar to them. The approach is depicted in Figure 1. The mandatory input of this approach is a to-be process model. Depending on the mining purpose, the to-be model could be a to-be process characteristic for a human with care level 2. Then the process is derived from documentation on care levels. Optionally, an as-is process model is taken as additional input. When the as-is state is not known, an empty trace is considered as as-is model. Based on this input a morphing is determined representing all valid and possible states from the as-is to the to-be process model. To determine the number of valid in-between states the process models are transformed in a textual representation of process trees, which we call a behavior-oriented trace (bt). An example bt would be [→a, x(b,c),d] meaning that a is executed in sequence, followed by b and c as alternatives and subsequently d follows sequentially. Then level recursion, deletion and insertion algorithms are applied to find possible clusters. In a nutshell, the level recursion

[1] Karlsruher Institut für Technologie, Institut für Angewandte Informatik und Formale Beschreibungsverfahren, Geb. 5.20, 76133 Karlsruhe, agnes.koschmider@kit.edu

algorithm decomposes behavior-oriented traces into array lists, attaches nesting depth to lists in order to improve indexing and intend to find n-gram items (at least 2-gram). The deletion and insertion algorithms delete or insert elements that are missing or unnecessary in *bt*s. In case that similarities or changes in event logs should be detected, then the Levenshtein distance in combination with a model-awareness notion are applied. The size of a cluster correlates with the group size (i.e., the more traces in a group, the larger the cluster). Our clustering approach can be evaluated based upon the “maximizing intra-cluster similarity and minimizing intercluster similarity" [WB13]. This means that a small distance between all elements within a cluster and a large distance between the groups should be produced. This is exactly the benefit of this clustering approach. All behavior-oriented traces on the root axis fulfill this requirement and thus make our clustering approach efficient. This clustering approach relies on the presentation of [Ko17]. The approach has been formalized and implemented in the meantime and performance analysis were conducted.

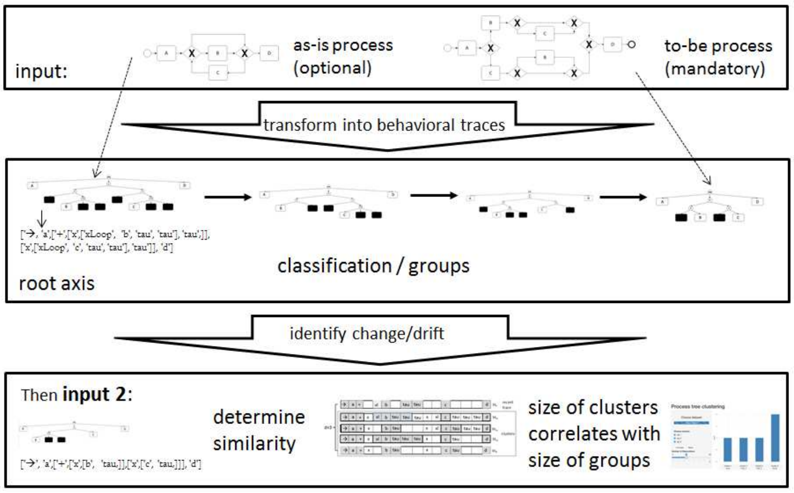
input:
as-is process
(optional)
to-be process
(mandatory)
transform into behavioral traces
['→, 'a',['+',['x',['xLoop', 'b', 'tau', 'tau'], 'tau',]],
['x',['xLoop', 'c', 'tau', 'tau'], 'tau']], 'd']
root axis
classification / groups
identify change/drift
Then input 2:
['→', 'a',['+',['x',[b', 'tau,]],['x',['c', 'tau,]]], 'd']
determine
similarity
size of clusters
correlates with
size of groups

Fig. 1.: Clustering technique based on the notion of behavior similarity and the number of clusters is determined beforehand. The size of a cluster correlated with the group size.

References

[Ko17] Koschmider, A.: Clustering Event Traces by Behavioral Similarity. ER Workshops, vol. 10651 of LNCS, Springer-Verlag, pp. 36-42, 2017.

[WB13] Weerdt, J.D., van den; Broucke, S.; Vanthienen, J.; Baesens, B.: Active trace clustering for improved process discovery. IEEE TKDE, 25(12), 2708-2720, 2013