Activity Mining by Global Trace Segmentation

Christian W. Günther, Anne Rozinat, and Wil M.P. van der Aalst

Information Systems Group, Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands
{c.w.gunther,a.rozinat,w.m.p.v.d.aalst}@tue.nl

Abstract. Process Mining is a technology for extracting non-trivial and useful information from execution logs. For example, there are many process mining techniques to automatically discover a process model describing the causal dependencies between activities. Unfortunately, the quality of a discovered process model strongly depends on the quality and suitability of the input data. For example, the logs of many real-life systems do not refer to the activities an analyst would have in mind, but are on a much more detailed level of abstraction. Trace segmentation attempts to group low-level events into clusters, which represent the execution of a higher-level activity in the (available or imagined) process meta-model. As a result, the simplified log can be used to discover better process models. This paper presents a new activity mining approach based on global trace segmentation. We also present an implementation of the approach, and we validate it using a real-life event log from ASML’s test process.

Keywords: Process Mining, Event Log Schema Transformation, Trace Segmentation.

1 Introduction

Process mining technology attempts to extract non-trivial and useful information about real-world processes from event logs recorded by IT systems that support these processes [1]. For example, there are many process mining techniques to automatically discover a process model describing the causal dependencies between activities [10,6]. Event logs are sets of traces, whereas a trace is a sequence of events referring to one particular instance of the process. An ideal event log for process mining analysis is well-structured and on an appropriate level of abstraction (e.g., one event in the log corresponds to the execution of one activity in the process). In many real-life situations, these requirements are, however, not fulfilled. Often, real event logs are recorded on a very low level of abstraction. Events in these logs are identifying miniscule activities within the system, which cannot be easily related to activities in the process model imagined by the analyst. It is not that these high-level activities are not represented in the event log at all, rather that their representation is scattered among many low-level events. This dissociation of activities makes it very hard for process analysts to correctly relate the observed behavior to any available, or imagined, process meta-model.

Trace segmentation is an event log schema transformation technique, which makes such low-level logs more understandable and easier to analyze. The fundamental idea of trace segmentation is illustrated in Figure [1]. The starting point is a low-level trace of
events (cf. bottom of Figure 1). Trace segmentation attempts to identify coherent subsequences of events within the trace, i.e., to “cut up” the trace into a number of event clusters. In the example in Figure 1 four clusters have been identified in the trace.

The rationale behind trace segmentation is that every cluster of low-level events is supposed to represent the execution of a higher-level activity in the (available or imagined) process meta-model. It is also important that these event clusters are properly categorized (i.e., clustered) into types of clusters. This allows for the discovery of corresponding activity types, which are supposed to result in a comparable sub-sequence of low-level events within and across different traces. With respect to the example in Figure 1 two cluster types A and B have been identified, each supported by two clusters. Clusters of type A consist of events from event classes A, B, or C, while clusters of type B are constituted by events of classes X, Y, Z, and W.

Trace segmentation has two main use cases. The first one is activity mining. In activity mining, trace segmentation is applied to elevate the log’s level of abstraction, i.e., to analyze the event log from a higher-level point of view. In the example in Figure 1, the trace would have been simplified to a sequence of four events (i.e., the clusters), from two event classes (i.e., the cluster types). The second use case is trace discovery. In trace discovery, the discovered event sub-sequences are interpreted as traces of a hidden sub process described by their cluster type. Regarding the example, the process type represented by cluster type A would have two traces A, B, A, C and A, B, C, B.

Both activity mining and trace discovery can be implemented by trace segmentation techniques. In the following section we propose a new global approach towards trace segmentation, which is based on the correlation between event classes.

2 Global Trace Segmentation Approach

As explained earlier, trace segmentation is based on the idea that subsequences of events, which are supposed to be the product of a higher-level activity, are identified. This approach focuses on the global correlation between event classes, i.e. types of events. From the co-occurrence of events in the log, we derive the relative correlation between their event classes.

Our approach for global trace segmentation can be outlined as follows. (1) We use the notion of a global event class correlation, describing how closely related the event