

logs. This approach deals with hierarchically structured workflow models that include the splits and joins.

Dongen and van der Aalst [7] specified a standard for storing event logs by introducing a data model and an XML format called MXML (Mining eXtensible Markup Language).

Weijters et al. [8] came up with an algorithm, called Heuristics Miner, which discovers main behavior registered in a noisy event log by including different threshold parameters in order to overcome two problems: noise and low frequency behavior.

Günther and van der Aalst [9] focused on existing problems in the traditional process mining methods when the processes are large and less-structured. In order to manage the problems, they built up a flexible approach based on their previous work [10]: Fuzzy Mining. Their approach analyzes, simplifies and visualizes mined process models based on the metrics significance and correlation of graph elements.

Fahland and van der Aalst [11] simplified discovered process models and demonstrated a post-processing approach to check the balance between overfitting and underfitting. They detailed the discovered process model in Petri net. Their approach can be combined with any process discovery method which generates Petri net.

Aleem et al. [12] proposed the comparison of different process mining methods in detail. The paper collects and shows all efficient and qualitative results of business process mining for researchers. Their article groups the process mining approaches to five sections: deterministic, heuristic, inductive, genetic and clustering-based mining approaches. Cheng and Kumar [13] suggested a method to eliminate noisy traces from event logs by way of building a classifier and applying classifier rules on event logs. They indicated that produced mined models from such preprocessed logs are superior on several evaluation metrics. Fahland and van der Aalst [14] investigated the problem of repairing discovered process model to align them to reality. They separated the event log into different sublogs of nonfitting traces to make conformance checking. Rovani et al. [17] demonstrated a methodology so as to analyze medical treatment processes through displaying how to apply process mining methods based on declarative models.

De Leoni et al. [16] suggested a framework to unify a number of approaches for correlation analysis. They tried to correlate different process characteristics related to different perspective. Mannhardt et al. [17] offered a process mining algorithm to control process conformance with respect to control flow, data dependencies, resource assignments and time constraints. Pika et al. [18] represented an approach which is based on the analysis of information about process executions recorded in event logs. They demonstrated a supporting tool to assess the overall risk of process and to forecast process outcomes.

Suriadi et al. [19] specified a set of data quality issues and suggested a patterns-based way to clean noisy event logs. Mitsyuk et al. [20] proposed a tool to produce event logs from Business Process Model and Notation (BPMN). They implemented script-based gateways and choice preferences to handle control flow. Bolt et al. [21] addressed the problem of

comparing different variants of the same process and observed differences in behavior and business rules. They used transition systems which were annotated with measurements to model behavior and to underline differences.

Yürek et al. [22] proposed a novel algorithm, named Interactive Process Miner (IPM), that has the capabilities of working on huge volume of event logs and managing the execution records of running process instances to generate process model in a short time with high precision and fitness values and holds three different features, including activity deletion, aggregation and addition operations on the process model to support a simulation environment for users.

Process mining tools named as Little Thumb, EMiT and ProM are developed by Van der Aalst and his team. Little Thumb can get workflow nets from noisy and uncompleted logs [10]. EMiT can convey workflow models with Petri nets [23]. ProM is a generic open-source framework for implementing process mining projects that contains many packages with many plug-ins [24].

Differently from the previous studies, this paper proposes a novel tool to analyze large volume of event logs and to handle the execution records of running process instances in a short time with low memory usage, and also support an interactive environment for process mining to give deep insights for event logs.

III. PROLAB: PROCESS MINING TOOL

There is a need for a software tool in order to analyze the process records and to extract the statistical information from event logs. This process mining tool should be able to work with low resource consumption on the large amount of event logs. An XML-based file called Mining eXtensible Markup Language (MXML) has been standardized to store event logs which is used in process mining. Thus, a general input format is provided for different process mining techniques and tools. In this study, our aim is to develop a process mining tool which is able to work with low resource consumption on huge amount of event logs and to provide an interactive environment for users.

A. Block Diagram of the Tool

The block diagram of the process mining tool we developed is shown in Figure 1. Event logs in MXML format are read using the file streaming method, so it is possible to read the event logs without loading the whole file into memory. The statistical information obtained from event logs during streaming is stored in a data structure. This data structure describes the summary information that will be used in process mining.

The process mining algorithm is executed on the data structure that holds the statistical information and a process model is created that represents the process flows. The generated process model is a summary information of the event logs. By visualizing the generated process model, this summary information is displayed to the user with a graphical interface. With the help of settings, users can filter both the event logs and change the visual appearance of the process model. These