Joint Master Thesis - Modelling of production expertise to extend the data-driven analysis of process models

Description

In operational processes such as production processes or order processing, the production expert takes a crucial role. Various process runs and sequences of activities occur in practice that are not target-oriented, for example when "production" precedes "engineering". For this reason, it is important to consider the knowledge of production experts for identifying process deviations. It is important to find out if predefined weaknesses occur in discovered processes.

For the integration of production expertise in data-based process analysis, declarative process mining, performance analyses, and deviation detection techniques in process mining need to be merged into one framework. As a result, human production expertise and data-based process discovery are combined. By declarative process mining, predefined rules of production experts are covered in discovering the process model. In the module for process model analysis, the deviation detection and performance analysis are performed. At the same time using the user input and declarative process mining techniques further deviations and weaknesses are discovered.

The goal of the thesis is an independent module for the automatic, data-based detection of process weaknesses. Based on predefined decision rules, this application examines event logs/process models of past process flows, as shown in Figure 1. The input is the event log of production processes. With the help of the modeled production knowledge, the module delivers both the discovered process models as output and the weaknesses identified in them. In this way, the knowledge of the production experts and the findings from the event log are combined to form a decision-related process analysis and interpretation. The production line expert’s knowledge can be the rules which are at an activity-flow level. They can be lists of forbidden activities and possible activity-flows (parallel activities). Therefore, the steps in the application are as follows:

1. Detect the ignored rules in the process.
2. Discovering the performance metrics and identifying the outliers (possible deviations).
3. Show the detected weakness:
   * E.g., in the process, activity “A” is performed before “B” which is not allowed according to the rules.
   * E.g., the detected performance metrics differences, e.g., activity “A” on average takes 2 minutes, for a specific case it took one hour.

A screenshot of a social media post

Description automatically generated

Goal>

**Input** : event log, activity-flows from production line experts

**Process** :

1. Convert activity flow to process model(expert-negative-model)

2. Convert the given event log into process model(current-model)

3. Compare the 2 models to check if the current-model is satisfying the expert-negative-model and if yes note down the results as weakness.

4. Do a performance analysis on the current model and identify the outliers and report them as weakness as-well.

**Output**: Discovered process model from the event log(current-model) + discovered process model from expert knowledge(expert-negative-model) + weakness identified in the discovered current-model.

**Challenges / questions to be answered**:

1.Can the activity flow be converted to a model/should the activity flow be kept as sequence of activities just like in an event log.

2. How can we compare 2 models / should the conformance checking of event log of experts be used to check on the current model.

3. Can we include performance conditions from the experts and include them while reporting the performance analysis of the current model

**Functions that are available in PM4PY for us to solve the problem**:

1. Event Log (objects.log.log.EventLog) ====For the current event log and convert the sequences of acivities into event log.

Assuming that the experts gives us the activity flow from beginning to end.

Sequence mining

CONFORMANCE CHECKING FOOTPRINT MATRIX

Conformance checking=>Token based replay and alignment