Conformance Checking 1

Group 7

Contact Person: Laura Genga, l.genga@tue.nl

Important Dates:

- 1st office hour: 21st October, 09:30-10:00h online via Teams, send your questions no later than 21st October, 08:00h
- 2nd office hour: 11th November, 09:30-10:00h online via Teams, send your questions and presentation draft no later than 11th November, 08:00h
- Hand-in presentation and implementation: 17th November, 23:59h
- Presentation: 21st November, 13:45-14:30h

Instructions for Deliverables:

- Implementation:
  - Programming language is Python 3.x, follow common best practices (check, e.g., https://peps.python.org/pep-0008/) and use ideally object-oriented structure
  - README.md with installation instructions and instructions how to run your code https://tilburgsciencehub.com/topics/collaborate-share/share-your-work/content-creation/readme-best-practices/
  - requirements.txt containing all necessary dependencies https://pip.pypa.io/en/stable/user_guide/
  - Deductions possible for too messy code (variable and function names not self-explanatory, structure unclear, etc.), wrong or incomplete information in README.md and/or requirements.txt
- Presentation: follow the template provided in Canvas

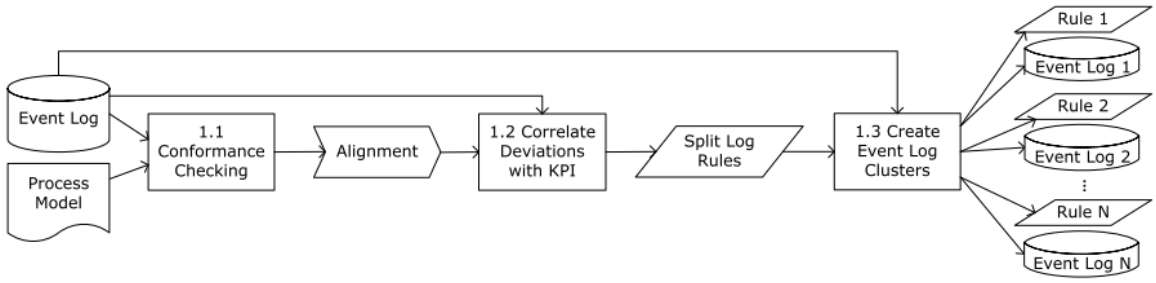
Topic Description:

Conformance checking techniques allow to compare process behaviors allowed by a normative process model with the behaviors logged into an event log during the actual process execution [1]. These techniques are commonly employed to pinpoint discrepancies between expected and actual behaviors and can return both local and global diagnostics. Multiple process perspectives are considered when generating process diagnostics, thus allowing to detect deviations related both to the control-flow (e.g., violations of ordering constraints among process activities) and violations of possible data guards stating additional conditions that must be satisfied to execute a

(set of) process activity (ies) (e.g., loan applications with an amount higher than 1000 might have to undergo to a different check than loan applications with lower amounts).

Previous studies highlighted that deviations might point out common workarounds actually linked to better process performance [2]. Being able to recognize these deviations is hence crucial to identify opportunities for process improvements.

Your goal for this assignment is to implement the "Deviation analysis" step reported in [2]. Your artefact must take the following inputs; a Petri net (in pnml format); an event log (in XES format). Your artefact must return i) a set of rules correlating each detected deviation with a predefined KPI, and ii) for each rule, a subset of traces from the original log which are compliant with the rule. The figure below illustrates the steps of the approach.



Your artefact must implement the following steps.

- First, you need to define a KPI of interest. For your analysis, you will consider the **throughput time** of each trace. Hence, the first step for your artefact is to compute the throughput time of each trace and add an additional log attribute for each trace corresponding to the computed value. You can, but it is not required, export the enriched log in a new XES file.
- Generate an alignment for each trace in the event log. From each alignment, generate an encoding for each trace. Trace features are: the number of model moves for activity a_i; the number of log moves for activity a_i. Therefore, two features are obtained for each process activity.
- Apply a decision tree on the encoded set of traces. Use your previously defined KPI as dependent variable. Apply commonly used strategies (e.g., hyperparameter optimization) to obtain the best performing classifier.
- Derive the set of classification rules from the decision tree and split the log accordingly.

Note that to implement the above-mentioned tasks you do not need to implement a fancy UI. Parameter selection from the command line is enough for your assignment.

You are expected to evaluate how well your approach

1. Generate the optimal alignment for each log trace
2. Find the right classification rules

For this you will have to manually develop a gold standard against which you can test your approach. Make use of at least precision and recall scores measuring the “how well”. Use further metrics if necessary.

Dataset(s) for Evaluation:

You can start with your own small artificial datasets for testing, but your approach must work in the end for the dataset provided with the assignment. An overview of the dataset is provided in the file “LogDescription.pdf”. Further information can be found at https://www.win.tue.nl/bpi/2017/challenge.html

The website also provides the best articles submitted to the corresponding competition, which provide an extensive analysis of the dataset.

Note; while developing the approach, you can use a sample of the event log to speed up the development, then using the whole dataset once you have a stable solution.

References:

[1] Song, Minseok, and Wil MP Van der Aalst. "Towards comprehensive support for organizational mining." *Decision support systems* 46.1 (2008): 300-317.

[2] Dees, Marcus, Massimiliano de Leoni, and Felix Mannhardt. "Enhancing process models to improve business performance: A methodology and case studies *OTM 2017 Conferences*. Springer International Publishing, 2017.