Process Mining: A Log Filtering API in Python

CSI 6900 Project Description, Summer 2022
Supervisor: Daniel Amyot
Student: **Dehui Yu** (300225499)

Context

Processes are running everywhere, and their number is rapidly increasing. The definition of a *process* is a series of progressive and interdependent steps by which an objective is attained [1]. Many organizations (such as schools, companies, government agencies, universities, etc.) now use information technology systems to support Business Processes [1]. We call a list of events recorded during the operation of information systems an *event log*. To better analyze these event logs, we will use the technology called **Process Mining (PM).**

Process mining is a family of techniques relating to the fields of data science and process management, and used to support the analysis of operational processes based on event logs. It can extract process maps (or models) from event logs collected by information systems [2]. Event logs can typically be used for three types of process mining tasks (see Figure 1).

1. *Process Discovery*: Process Discovery is a technique for building process models from event logs without any prior knowledge. If the event log contains resource information, resource-related models (such as social networks) can be discovered [1].
2. *Conformance Checking*: It is a technique that compares an existing process model to an event log from that process. Conformance checking is used to verify that the real process (recorded in the log) is consistent with the process model [3].
3. *Enhancement*: improves and extends an existing process model based on insight generated by process mining [1].

We will focus on Process Discovery in this project.

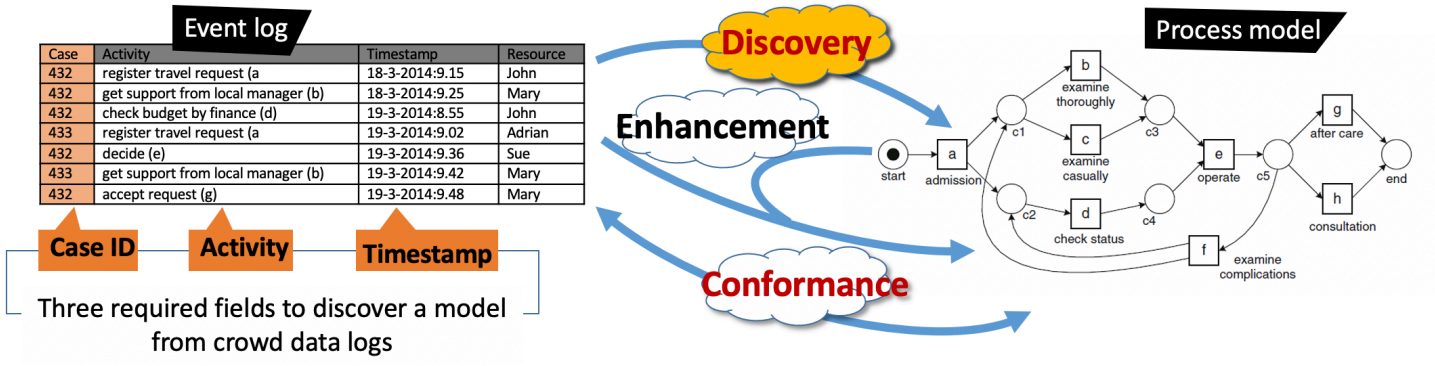

Fig1. Overview of process mining [2]

Unfortunately, event logs from Information Systems are usually not handled directly out-of-the-box by process mining tools. To be effective, all PM approaches include a **log filtering** step to clean and simplify the event logs before feeding them to mining algorithms and process visualization engines. A recent thesis [4] proposed a new API with supporting

functions that enable people to create useful and simple scripts that automate event log filtering. Below is expected to be completed APIs:

	Row	Column	Trace	Other
Add		concatenateColumns() eventIsRepeated()		
Remove	keepFirstEvent() keepLastEvent() filter() removeEvents() LowFreqency() deleteAllEvents()		deleteTraceLengthLessThan() deleteTruncatedTracesStart() deleteTruncatedTracesEnd() deleteTracesWithTimeLess()	
Modify	CleanHeaders() arrangeRows() mergeRows()			
Other				readCSV() writeCSV()

A current implementation of this API in *R* already exists. The project aims to research, design and develop a solution that would support a new *Python* version of this API for log filtering. Additional functions may also be considered along the way, e.g., to better support Robotic Process Automation.

PM4Py is a python package that supports (state-of-the-art) process mining algorithms in Python [5]. It is completely open source and intended to be used in both academia and industry projects. PM4Py implements the latest, most useful, and extensively tested methods of process mining.[6] PM4Py is a product of the Fraunhofer Institute for Applied Information Technology, in Germany. This project will also use this popular open-source Processing Mining tool to process the event log and integrate it with our Python-based implementation of the API.

Project Objectives

This Intensive Graduate Project in Computer Science has the following objectives:

1. A new Python API for log filtering. This will be implemented by selecting existing libraries to support these functions, as well as developing the missing ones.
2. Checking the performance of that library compared to the R-based one
3. Integrating this to a popular open-source Process Mining tool written in Python (PM4PY)

Learning Objectives

- Learn the design ideas and syntaxes of the R and Python programming languages.
- Learn the difficulties in supporting process mining effectively.
- Learn how to use a common process mining tool (PM4Py).
- Learn how to search existing libraries and select those that can best support an API, taking licensing into consideration.
- Learn how to design, improve, implement, and test an evolving Application Programming Interface.
- Learn how to analyze and report on the performance of a technology.
- Learn how to produce reliable scripts that combine multiple technologies (our new API, and PM4PY).

Deliverables and Marking Scheme

- 5%: Presentation and demo of the PM4Py tool
- 20%: First version of the API, with code inspection
- 20%: Second version of the API, with code inspection
- 15%: Third version of the API, with code inspection
- 15%: Testing and performance evaluation, with a comparison with the R-based implementation
- 15%: Integration with a Python-based process mining tool (PM4PY)
- 10%: Code, tests, and design+user documentation on GitHub
- Bonus 5%: Participation to the drafting of a scientific publication on the results
- Bonus 5%: Provide an additional integration of the API with a second PM tool (e.g., Celonis or Apromore)

Study Plan (290 hours)

Week	Activity	Hours
1	Familiarization with the project requirements; setup of GitHub and of IDEs, paper reading	**20**
2	Learn R programming language, understand its concepts, basic syntax, and IDE	**20**
3	Learn how to use the PM4PY tool and mine process models from existing data sets	**30**
4	Investigate the existing filtering API based on the R language, and test it on existing data sets	**20**
5	Search and select existing Python libraries that can support defined API functions	**20**
6	Provide a first version of the Python API (possibly incomplete) based on existing libraries	**20**
7	Program and test missing API functions	**30**
8	Provide a second version of the Python API (likely complete)	**20**
9	Test and evaluate the performance based on the thesis datasets and R-based results	**30**
10	Integrate scripting with PM4PY	**20**
11	Investigate API extensions for supporting Robotic Process Automation	**20**
12	Provide a third version of the Python API (likely complete), with test results	**20**
13	Presentation of results, and completion of online documentation	**20**

References

1. van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Verlag, Berlin (ISBN 978-3-642-19344-6).
2. van der Aalst, Wil (2016). *Process Mining: Data Science in Action*. 2nd edition. Springer.
3. Carmona, J., van Dongen, B.F., Solti, A., Weidlich, M. (2018). Conformance Checking: Relating Processes and Models. Springer Verlag, Berlin (ISBN 978-3-319-99413-0).
4. El-Gharib, N.M. (2019) *Using Process Mining Technology to Understand User Behavior in SaaS Applications*. MSc thesis, University of Ottawa, Canada. Online: https://ruor.uottawa.ca/bitstream/10393/39963/1/El-Gharib_Najah_Mary_2019_thesis.pdf
5. Fraunhofer Institute (2022) PM4Py: State-of-the-art-process mining in Python. Online: https://pm4py.fit.fraunhofer.de/
6. Drakoulogkonas, Panagiotis & Apostolou, Dimitris. (2021). On the Selection of Process Mining Tools. *Electronics*. 10. 451. 10.3390/electronics10040451.