Project Initiation

Label Refinement by Behavioral Similarity

Document owners:
Bianka Bakullari
Christopher Beine
Nicole Ventsch
Juan

Last edited: April 21, 2019

1 Overview **1**

2 Business Case **1**
2.1 Scope . . . 1
2.2 Assumptions . . . 1
2.3 Key Benefits . . . 2

3 Feasibility study **2**
3.1 Theoretical point of view . . . 2
3.2 Technical point of view . . . 2
3.3 Risks and mitigations . . . 3
3.3.1 Project management risks . . . 3
3.3.2 Technical risks . . . 3

4 Project Plan **3**
4.1 Milestones . . . 3
4.2 Deliverables . . . 6
4.3 Timetable . . . 6

5 Project Team **6**
5.1 Competences . . . 7
5.2 Roles . . . 7

6 Project office **8**
References 8

1 Overview

Many processes involve carrying out an action multiple times. An example for this would be an online shop in which you first have to pay a registration fee before ordering an item and paying it. This process contains the event "payment" twice, but in different contexts, so that the payments are actually two different tasks. In the context of analysing processes, the event logs usually only contain the event names, so that the "payment" actions would be treated as the same task and loops would be induced in the resulting models. However, these loops do not match the actual process, which is the issue this project addresses.
These imprecise logs should be refined based on the structural contexts of the events. We want to refine the logs without any filtering. Moreover, we want to allow an interactive change of the thresholds used to refine the labels since this can differ for every log and we have no knowledge of the correctness of the refined log in general. All of this should be done by creating a Python-based web service in this project.

2 Business Case

Based on the situation explained in the overview, we now want to discuss the business case and explore the project's potential. We will do this by first defining the scope of the project, then the assumptions we take and finally we will consider the advantages that are gained by carrying out this project.

2.1 Scope

During the project, we will create both code and documentations. Thus, the scope will be divided into these two aspects:
Documentation:

- create a Project Initiation Document
- realize the Requirements Specification document
- develop and describe the design of the interface
- design the algorithm structure by stating the usage of classes as well as the inputs and outputs for the functions that are used
- Software Documentation during milestones

Implementation:

- set up a Web Service based on Python that uses the label refinement algorithm proposed by Xixi Lu, et al.
- create a user interface that allows the users to upload the original event log, set thresholds and imprecise label scope and finally download the refined event log

For the realization of our project some background in Process Mining is needed, as well as programming knowledge. Moreover, the documentation of both organizational aspects as well as software will follow the software development lifecycle where each part has to be submitted before the corresponding due date stated in the section "Project Plan - Milestones".

2.2 Assumptions

In this project we will assume that an event log is given by the user, i.e., data that contains at least the attributes "id", "time stamp" and "activity name". Moreover, we will assume that these event logs are given in the standard XES format. Since the set of candidates for imprecise labels is given by the user, we assume that the quality of the processes applied on the refined event log also depends on the domain knowledge of the user who picks the candidate labels. Our algorithm is a pre-processing step in itself and it does not include other preprocessing mechanisms like filtering. In this context, instead of considering imprecise labels as noise and filtering them out, we change the labels based on the patterns present in the data.

2.3 Key Benefits

With this project we mainly aim at improving given event logs by making them more precise, so that the subsequent analysis will be more accurate. Our Label Refinement algorithm gives data scientists the opportunity to apply the process discovery algorithms to both original and refined event logs, leading to more alternative solutions from which the best can be chosen. More correct models enable gaining better insights from the data, which on the other hand gives the data scientists the opportunity to take the best measures in order to optimize the processes and that way reduce the company's expenses while increasing its efficiency. By designing an interface that allows the users to upload an event log, to set the thresholds and to download the modified event log, carrying out this project will save the data analysts a lot of time which would be needed to refine the log themselves and also help them find better models systematically. By providing a web service, the software can be used on demand and there are only minor requirements for the user's hardware which avoids the necessity to stick to a given programming language or software.

3 Feasibility study

In this section we give a description of our project by explaining why the task is realizable from both theoretical and technical point of view. We also try to take potential risks into account and propose countermeasures.

3.1 Theoretical point of view

From a theoretical point of view, we view an event log for which we suspect to have different tasks being handled as the same activity as *imprecise*. In reality, each event in the data is a unique occurrence over time. By defining a *labeling function* which maps these events to a finite set of activity names, also called *labels*, we obtain event logs having common activities across traces and within traces. A problem arises when similar events happening in different contexts are assigned the same label. Applying process discovery algorithms to such event logs may lead to process models being imprecise, misleading or even incorrect.
There has already been some research investigating this problem. Some of the approaches include using additional domain knowledge to correct the labeling, or trace clustering to do relabeling according to the different variants.
In our project, we focus on the algorithm developed, investigated and explained in [1], which given an event log, refines the labels of events based only on the context and patterns present in this event log. The approach consists of three main steps: 1)detecting activities which need refined labels, do relabeling 2)accross dissimilar traces and then 3)within traces. Here we assume that the set of candidate activities needing relabeling is given and the user should have the possibility to set the thresholds affecting the relabeling in steps 2) and 3).
Since the logs have a finite set of activities leaving room for only a reduced number of computations needed for the cost function in step 2) and the fact that this algorithm has already been successfully implemented in ProM yielding good results, we conclude that the project is feasible from a theoretical point of view.

3.2 Technical point of view

As stated in [1], the algorithm has already been implemented in ProM, where controlled experiments have been made to test the performance of the algorithm based on event logs containing different patterns. There has also been a real life case study involving an event log with data concerning healthcare which was provided by Maastricht University Medical Center (MUMC+), a large academic hospital in the Netherlands. In the log containing 1039 cases and 6213 events, the activities *surgery* and *consultation* were imprecise. After relabeling, the resulting model reflected the behaviour which was described by the domain experts, according to whom the acitivity *surgery* for example took place many times during the treatment of certain patients.
For our project we will use following open source tools:

- Python 3.7.2 as back-end programming language
- pm4py python library as process mining toolkit
- NetworkX pyhton library for graph manipulation
- Flask 1.0.2 as web application framework

- JavaScript as Front-end development language (Standard ECMAScript 2018)

We will seperate the application into two main parts the back-end which is responsible for the algorithm and API design and the front-end to provide a user friend interface.

Python is used as back-end programming language to implement the algorithm. Additionally, we will use Flask as a lightweight webapplication framework. It provides a quick start and is easily customizable via plug-ins. It is also used in enterprise applications like Netflix Lemu [?].

In particular we will make use of the .pm4py library which is a joint effort of the Process Mining group at the Fraunhofer FIT and the Chair of RWTH Process and Data Science group as it supports many process mining algorithms and performance measures in Python.

The frontend is for visualization only. It is created with HTML5 and JavaScript to access the API and to perform ui logic. Thus, presentation and business logic can be easily separated and replaced.

The final web application will be optimized for Google Chrome (Version ≥ 65) and Firefox (Version ≥ 60).

3.3 Risks and mitigations

In the following we try to estimate the potential problems and risks that may arise during our project and propose ways to be prepared for or overcome them. We take into consideration complications concerning the organisational and management aspects, as well as those arising when implementing the algorithm.

3.3.1 Project management risks

Risks	**Mitigations**
Misunderstanding of tasks	Do regular discussions, meetings in the group and with the project supervisor
Unmatching schedules of the team members	Arrange meetings in advance, notify other members for any change
Acquiring new skills under time pressure	Assign tasks based on individual strengths to increase efficiency

3.3.2 Technical risks

Risks	**Mitigations**
Inconsistencies in programming styles	Explain the code to other members, write useful comments while coding
Unclear performance measures	Test code on small event logs, use automatically generated event logs to experiment
Software components do not work as a whole	Put emphasis on the design step

4 Project Plan

4.1 Milestones

The project starts on the 09/04/2019 and ends on the 08/07/2019 and is divided into nine milestones. The project is managed with a scrum-oriented approach where each milestone represents a sprint. The project team organizes the required tasks during each sprint and visualize the current project status via a dashboard.

Table 1: Overview Milestones

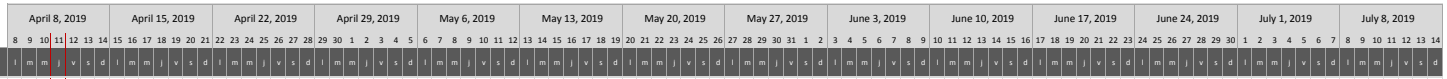
ID	Milestone	Description	Deadline
1	Project Initiation document	The Project Initiation Document provides all of the key information required to start and run the project. This includes the project description, business case, feasibility study and a project team presentation.	19/04/2019
2	Requirements Specification document	The Requirements Specification document contains functional and none functional requirements such as a set of use cases to describe the system interactions.	29/04/2019
3	Design Analysis and dummy P.o.C.	The final document is a description about the planned system architectural background and a proof of concept visualizing the main UI components.	13/05/2019
4	Sprint 1 code and documentation	In this sprint the import module and a static frontend will be implemented.	24/05/2019
5	Sprint 2 code and documentation	In this sprint the algorithm will be implemented.	07/06/2019
6	Sprint 3 code and documentation	IN this sprint the final user interface and visualization will be implemented.	21/06/2019
7	Testing, assessment and deployment	The application is checked for accuracy and should be aviable for use.	01/07/2019
8	Final report on the project	The final report provides an overview about the project course and the result.	08/07/2019

Label Refinement based on Behavioral Similarity

Project Start: 11/04/19

Display Week: 1

TASK	START	END	DAYS
Project Initiation			
Set up the Project Office	11/04/19	12/04/19	2
Develop a Business Case	11/04/19	16/04/19	6
Undertake a Feasibility Study	12/04/19	17/04/19	6
Establish the Project Charter	15/04/19	19/04/19	5
Appoint the Project Team	16/04/19	21/04/19	6
PI document elaboration	11/04/19	21/04/19	11
Requirements Specification			
Requirements analysis	22/04/19	24/04/19	3
Use case analysis	24/04/19	26/04/19	3
Requirements review	26/04/19	28/04/19	3
RS document elaboration	22/04/19	28/04/19	7
Design Analysis & dummy P.o.C.			
Architectural design	29/04/19	01/05/19	3
High-level design	01/05/19	03/05/19	3
Detailed design	04/05/19	12/05/19	9
Frontend mockup	05/05/19	09/05/19	5
Documentation review	29/04/19	12/05/19	14
Sprint 1			
Web Service set up	13/05/19	15/05/19	3
Static frontend	13/05/19	17/05/19	5
File loader module	13/05/19	17/05/19	5
Unit test	17/05/19	18/05/19	2
Module integration I	18/05/19	24/05/19	7
Documentation review	13/05/19	24/05/19	12
Sprint 2			
Imprecise label detetion module	24/05/19	29/05/19	6
Cost function module	24/05/19	31/05/19	8
Mapping module	24/05/19	31/05/19	8
Unit test	01/06/19	03/06/19	3
Module integration II	04/06/19	07/06/19	4
Documentation review	24/05/19	07/06/19	15
Sprint 3			
Horizontal/Vertical refinement module	10/06/19	17/06/19	8
File exporter module	10/06/19	13/06/19	4
Petrinet visualization module	10/06/19	14/06/19	5
Unit test	15/06/19	17/06/19	3
Module integration III	17/06/19	20/06/19	4
Documentation review	10/06/19	21/06/19	12
Testing, assessment and deployment			
Integration testing	22/06/19	25/06/19	4
Internal project assessment	26/06/19	27/06/19	2
Deployment	28/06/19	29/06/19	2
Final documentation	22/06/19	01/07/19	10



4.2 Deliverables

With each milestones various deliverables are created to monitor, document and verify the docuemnt progress.
1 Project Initiation document:

- Finale project initiation document with key information about the project

2 Requirements Specification document:

- Requiredments Specification document with functional und none functional requirements
- Use case analysis

3 Design analysis and dummy P.o.C:

- System and software architecture documentation
- Frontend mockup

4 Sprint 1 code and documentation

- Unit Test protocols
- Code documentation

5 Sprint 2 code and documentation

- Unit Test protocols
- Code documentation

6 Sprint 3 code and documentation

- Unit Test protocols
- Code documentation

@all: should we write tests during the implementation? Could possibly save us a lot of trouble, but we have to select a test framework. Otherwise we can remove the Unit Test protocols
7 Testing, assessment and deployment

- Testprotocols
- Server configuration
- Web API
- Webapplication

8 Final report on the project

- Final report

@all: Any other documents, reports, protocols or code artefacts required?

4.3 Timetable

5 Project Team

In the following we focus on the individual strengths of the members of the team and specify the responsibilities with respect to the milestones of the project.

5.1 Competences

In this section we introduce the members of our team by shortly describing their knowledge and competences which will be useful to carry out the task successfully.

Nicole Ventsch

Nicole Ventsch is a Master student at RWTH studying Mathematics and Data Science in parallel. She is studying both subjects in her second semester of the master's degree. Moreover, she works as a student assistant in the field of Data Analytics / Business Intelligence. Due to her background in mathematics, she has a good understanding of theoretical foundations. Moreover, she is very interested in Data Science and already took many courses in that area. Since she took the course "Introduction to Data Science", she has also worked with Python before. Though she has a strong theoretical background, she has never worked on user interfaces or with web services, so that this aspect of the project will be new to her. Additionally to that, she is not used to the software development Lifecycle, so that executing this project will include many new experiences to her.

Juan Garza

Juan Garza is a student at the RWTH University. He is currently in his 4th semester of studies towards a master's degree in Computer Science. During his studies, he deepened his knowledge of process mining by attending the lecture "Business Process intelligence". Moreover, he carried out a seminar on "Selected Topics in Process Mining". He is familiar with Java and Python as programming languages. Besides school projects, he has no programming experience in "real world" environments which may represent a difficulty for him.

Bianka Bakullari

Bianka Bakullari is a Master student at RWTH studying Computer Science. She has a good theoretical background on the field of Process Mining after having attented the lectures "Business Process Intelligence" and "Introduction to Data Science". Currently she is attending the "Advanced Process Mining" lecture. She has gathered some experience working with Python where in most cases pre-processing and analysis was made on concrete event data, therefore a challenge of this project is implementing an algorithm that works on any given event log efficiently. Since she has never worked with web applications before so the execution of this step will be a new experience for her.

Christopher Beine

Christopher Beine is a 4th semester Bachelor student at the RWTH Aachen University. Before his studies, he made an apprenticeship as a Computer Science Expert Subject Area: Software Development by Phoenix Contact GmbH & Co. KG where he works as a full-stack developer. He supervised several software projects within the company and is currently developing a web application for planning and visualizing the strategic orientation. In the field of data science, he deepens his knowledge with online courses such as the "Machine Learning" from Standford University on Coursera. He has practical experience with C#, PHP, JavaScript, the software development lifecycle and the most common web technologies.

5.2 Roles

We emphasize that all of the team members are supposed to put active effort on each of the milestones of the project. However, we assign a team member to have main responsibility for each step in order to ensure that the deliverables have sufficient quality and nothing is missing.

ID	Milestone	Chief Responsibility	Contact
1	Project Initiation document	Bianka Bakullari	bianka.bakullari@gmail.com
2	Requirements Specification document	Nicole Ventsch	nicole.ventsch@rwth-aachen.de
3	Design Analysis and dummy P.o.C.	Christopher Beine	christopher.beine@rwth-aachen.de
4	Sprint 1 code and documentation	Juan Garza	juan.garza@rwth-aachen.de
5	Sprint 2 code and documentation	Bianka Bakullari	bianka.bakullari@gmail.com
6	Sprint 3 code and documentation	Christopher Beine	christopher.beine@rwth-aachen.de
7	Testing, assessment and deployment	Juan Garza	juan.garza@rwth-aachen.de
8	Final report on the project	Nicole Ventsch	nicole.ventsch@rwth-aachen.de

6 Project office

In order to better be able to keep track of our progress, we use the Trello dashboard to divide our tasks into planned, running and completed ones. Trello also gives the opportunity to write descriptions and comments on each task, so that everyone is able to specify certain things needed to be considered or problems having arised. For our deliverables we use Git, as this makes it able to keep track of any changes and updates to our documentation or code. All members and the instructors have access to both Trello account and Git repository.

References

[1] Lu, Xixi, et al. "Handling duplicated tasks in process discovery by refining event labels." International Conference on Business Process Management. Springer, Cham, 2016.