Rheinisch-Westfälische Technische Hochschule Aachen
Chair of Process and Data Science
Prof. Dr. Wil van der Aalst

Process Conformance Checking in Python in SS 2021

# Alignments on NFA(s) in Micropython

*Group:3*

Requirements Engineering

13.5.2021

supervisor: Alessandro Berti

# Contents

# List of Figures

# 1 Requirements Elicitation and Development

Requirement gathering is the most critical phase that has a direct impact on the success or failure of any project. The traditional requirement gathering approach comprises of taking interviews from the direct stakeholder i.e. the client and merging it with your perspective of what the end user wants. In a world where now even the end user is mostly unsure of his/her wants, this approach is being replaced by design thinking techniques. We have implemented these techniques in our project to not only rely on the apparent wants of our client and end users but also to inject even the needs they are not aware of at this point in time. We started off with creating the territory maps to define user personas i.e. who will be the main users of our product and how will they interact with it. We then conducted interviews with the client first and then with randomly selected sample from those defined user personas to gain more insights on the problem and the possible solutions. Both open and close ended questions were asked to ensure every aspect is covered in those sessions. We then created high level customer journeys based on the information collected so far to visualize what each user of our product will go through to meet the required objectives. This highlighted loopholes and challenges throughout the product pipeline that otherwise would not have been known. Details of the identified stakeholders and their requirements are as follows:

## 1.1 Stakeholders

We considered the view point of following stakeholders while gathering the requirements of our product.

- Business Customer Company
- Developer Customer Requirements that will be using / extending the library in the future
- Internal Developers

## 1.2 Raw Requirements

### 1.2.1 Business Customer Requirements

- The software runs as expected - secured by tests
- The software does not depend on any other library
- The results of the software should be reproducible
- The software should compute the following results
  - A process model (NFA) given a regex
  - One of the optimal alignments given a trace of an event log and a regular expression that describes a NFA.
  - Calculate various alignments for a given trace and returning the optimal one by checking the fitness
  - Cost of the alignments should be returned

- The software should be able to understand and import and export the following formats
    - Regex as a string
    - Alignments
    - Event logs (standard: XES)
- The alignment cost function defaults to the one where a sync move cost 0 and all other cost 1
- A cost function can be specified
- The software should have a user guide documentation
- The software should come with a open-source license
- Simple user journey and a user-friendly design so that any layman can use the product easily
- Minimum possible response time

### 1.2.2 Developer Customer Requirements

- Regular expressions can be converted into NFA models
- Alignments should be in a format that can be used for further analysis
- The usage paradigm of the software should be aligned with PM4PY
- There should be a documentation on how to use the software
- There should be a implementation documentation
- Module base development so that it can be extended easy in the future

### 1.2.3 Internal Developer Requirements

- The software should be easy to maintain
- The software should be extensible
- The used file format should match the one used in PM4PY
- The code should be easy to maintain and have a high quality
- The memory usage should be kept low

### 1.2.4 System Requirements

- The software should run on Python
- The software needs no software requirements in the from of dependencies
- The minimum Python Version is 3.6

## 1.3 Requirement Analysis

## 1.4 Allocation and Flow Down of Requirements

### 1.4.1 High Level

- **H1** The interface should be simple.
- **H2** The response time should be minimized.
- **H3** The software should have no dependency.
- **H4** The software should have low hardware and software requirements.
- **H5** Proper User guide should be provided with the software.
- **H6** Proper documentation should be done for future reference.
- **H7** The code base should be efficient.
- **H8** Multiple operating systems should be supported.
- **H9** The software should return the most optimal alignment using NFA.
- **H10** The results should be an optimal alignment.
- **H11** The software should generate and operate on NFAs.
- **H12** Errors should be handled properly.
- **H13** The software should be extensible for future additions.
- **H14** Dynamic cost function should be supported.

### 1.4.2 Intermediate Level

- **I1** Interfaces should be designed keeping in view optimal User experience. **(H1, H2)**
- **I2** A consistent naming convention should be used. **(H1, H2)**
- **I3** A consistent coding approach with techniques used in the PM4PY library should be used. **(H1, H2)**
- **I4** Software should not be dependent on other libraries. **(H3)**
- **I5** Software should support at least Python versions 3.6. **(H4)**
- **I6** Software should be able to run on various operating systems. **(H4, H8)**
- **I7** All possible ambiguities should be made clear in the provided user guide. **(H5)**
- **I8** All aspects should be well documented. **(H6)**
- **I9** The entire workflow should be documented for ease in re-usability. **(H6)**
- **I10** The library should be easy to use, understand and should have consistent error handling techniques. **(H7,H14)**

- **I11** Internally the calculation of various alignments for a given trace should be supported. One optimal alignment should be returned. **(H9)**
- **I12** The optimal alignment should be computed based on the results of the cost function. **(H9, H10)**
- **I13** The library can reuse existing python standard libraries. **(H7)**
- **I14** The software should be able to export and import event logs and alignments. **(H11)**
- **I15** A conversion from regex to NFAs should be supported. **(H11)**
- **I16** Testing standards should be applied. **(H12)**
- **I17** A modular approach for software should be implemented. **(H13)**
- **I18** Software should provide user configurable cost function **(H14)**

### 1.4.3 Low Level

- **L1** Entire Customer Journey funnels should be designed in a way that each actor of the system has top quality user experience so that they remain loyal to our product. **(I1, I2)**
- **L2** User should be allowed to define their own cost function if required.**(I18)**
- **L3** The test suite should be executed on different OS before release.**(I6, I16)**
- **L4** A documentation that describes how to get started using the software should be given. **(I7)**
- **L5** A documentation that describes the functionality of the components should be given.**(I8, I9)**
- **L6** All the steps should be clearly written down for easy on-boarding of people who will be using or extending the work done by our team. **(I7, I8, I9)**
- **L7** Decoupled components should be used to ease extension in the future. **(I13, I17)**
- **L8** Event log in the XES format should be imported. **(I14)**
- **L9** The software should be implemented in Python. **(I3)**
- **L10** No external libraries should be used. **(I4)**
- **L11** One should be able to save the calculated alignments. **(I14)**
- **L12** The optimal alignment should be the one with lowest value for the cost function. **(I12, I18)**
- **L13** The test suit should check for problems concerning the responsiveness.**(I16)**
- **L14** The test suit should execute unit tests.**(I16)**

- **L15** The given regex input should be validated.**(I15)**
- **L16** Utilize the Python exception handling schema. **(I10)**
- **L17** Common XES log import functions from PM4PY should be used.**(I14)**
- **L18** The NFAs should be generated based on the given regex.**(I15)**
- **L19** Alignments should be calculated based on the given event log and a given NFA.**(I11)**
- **L20** The most optimal calculated alignment in terms of the alignment cost should be returned.**(I11, I18)**
- **L21** Support for Mac OS, Windows and Linux should be provided.**(I6)**

## 2 Functional Model

A key aspect of requirement engineering Is the construction of a functional model. In requirement engineering, a functional model is a graphical representation of a function as a process model representing the input, how the input is processed, and the output. Our system consists of two main actors namely:

- The client application: this is the application that the clients will use4 to access our library and make calls to our library.
- Our library: this is the library that we are going to implement and it is the component that the client application directly communicates with and which returns the optimal alignment and the accuracy and it doesn't make any calls to external libraries.

## 2.1 The Model

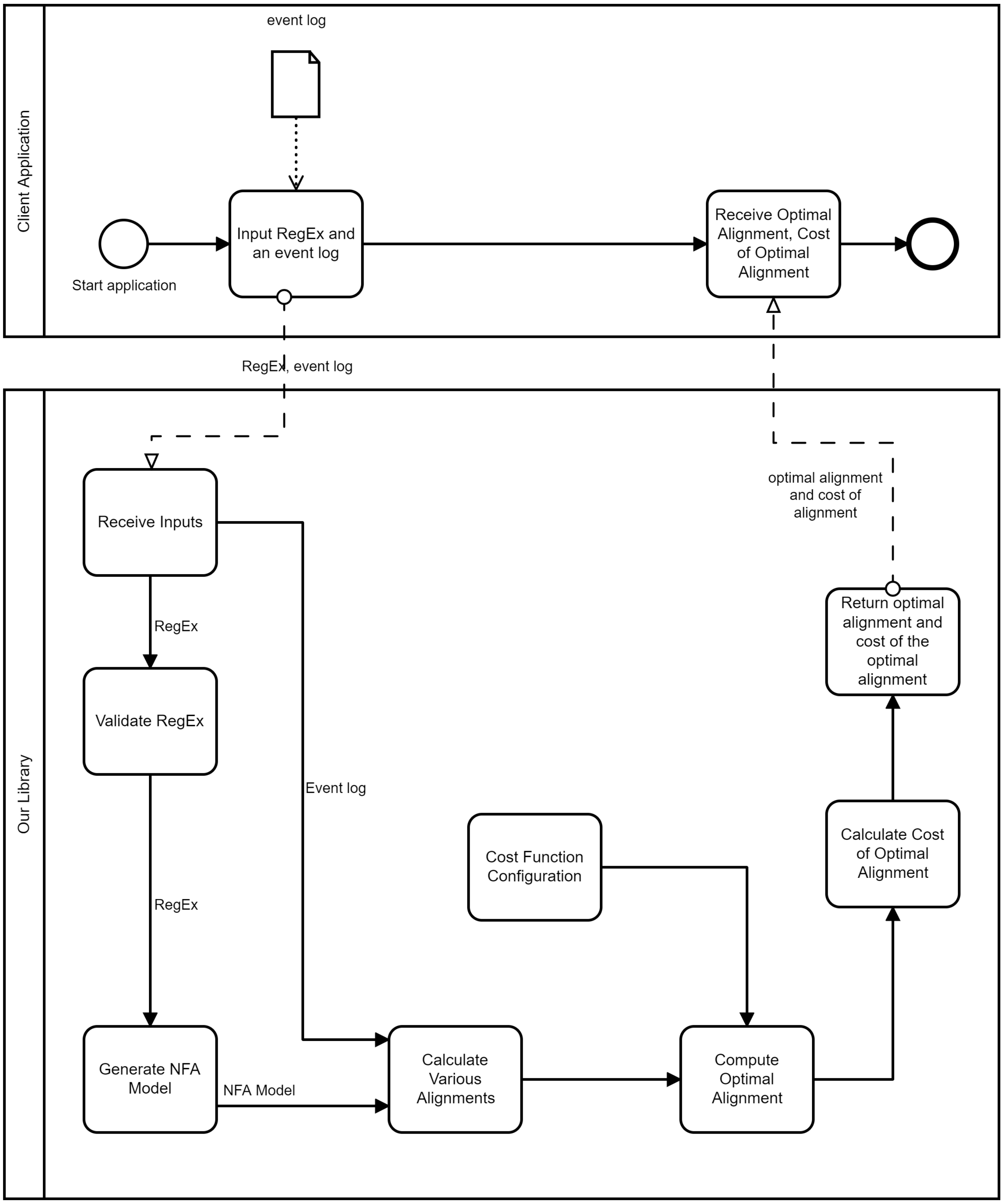
Client Application
event log
Start application
Input RegEx and an event log
Receive Optimal Alignment, Cost of Optimal Alignment
RegEx, event log
Our Library
Receive Inputs
RegEx
Validate RegEx
Event log
RegEx
Generate NFA Model
NFA Model
Calculate Various Alignments
Cost Function Configuration
Compute Optimal Alignment
Calculate Cost of Optimal Alignment
Return optimal alignment and cost of the optimal alignment
optimal alignment and cost of alignment

Figure 1: This model shows the process of alignments on NFAs(Cawemo.com)

## 3 Validation and Verification of Requirements

It is important to involve other people than just the project team, especially the stakeholders, in the requirement evaluation in order to improve the quality of the requirements. Therefore we evaluated the initial set of requirements with Mr. Berti, the contact person of our customer company. We decided on a few changes in the requirements which had been updated as follows:

- H2 The response time should be minimized
  - Changed to be measurable.
- I5 Software should support at least Python version 3.6
  - Changed to be more precise so the specific version was added.
- I11- Only one optimal alignment with minimum cost would be returned
  - Changed to be more concrete.

## 4 Requirement Management Tools

Requirement gathering and requirements management are the most vital parts of Software Development Lifecycle. The success of the project depends heavily on this phase. If this phase is not done with honesty, the foundation of the whole Project would be very fragile. Unfortunately, this is not a static part of the project, rather it evolves dynamically throughout the SDLC. New requirements can arrive at any stage of the SDLC. Similarly, requirements change overtime, some gets prioritized, others become irrelevant, even some requirements change completely with time. All these dynamically changing requirements must be communicated to the whole team and every team member must be aware and up to date with these changes. Handling this, is not easy when done manually, hence we need some state-of-the-art Requirement Management tools. Requirement management has following four Industrial Best Practices[7],

- Good Requirements
- Collaboration and Buy-In
- Trace-ability and Change Management
- Quality Assurance

We have evaluated some of the tools that can help us managing our requirements.

### 4.1 Orcanos

Orcanos[6] is a compliance management SaaS that efficiently handles the Design Control, Risk Management, and Quality Management Assurance of different regulated companies. Orcanos collaborative environment has proven to speed up our customers delivery, while reducing overall failure cost and risk, and improve quality significantly. The great power of Orcanos is in the integration of Research and Development, Quality, Delivery and Regulation. Along with these, it emphasizes on visualization and has an interactive and easy to use interface. This tool offers end-to-end trace-ability, test management, and simple collaboration tools such as chat and alerts. One interesting feature of this tool is DocGen, which helps to generate documentation for each stage of the project in the SDLC.

### 4.2 Jama Software

Jama Software[8]is the definitive system of record and action for the development of project. The company's efficient and interactive solution helps enterprises accelerate development time, reduce risk, slash complexity, and verify and validate regulatory compliance. The software has dedicated features for Reviews, Team Collaboration, Impact analysis, Risk management and Compliance Regulation. It has an online web hosted environment which makes it accessible to all. The tool makes it easier for the team to efficiently communicate Requirements, updates, progress, and Goals.

### 4.3 Pearls

PEARLS [9]is one of the most affordable online requirement management tools which has been designed to help all the business analysis requirements with efficiency and easiness. The tool fits with Software development life cycle models like Waterfall and Agile models. It also supports Hybrid SDLC model. The Tool has many useful features like Activity Notifications, Comments, User Role, Backlog etc. It enables users to draft Project artifacts like Use Case Document, Requirement Specification Document etc.

### 4.4 Caliber

It is one of the best available tools for Visualization of the requirements. It has been designed by keeping Human Computer Interaction as the most focused item[10]. A simple yet interactive Drag and drop interface makes it very easy to use. Users can attach attachments to the requirements. But the major drawback of this tool is that it is a desktop application with no Cloud hosted application which means that every user has to setup this tool at their personal Systems.

### 4.5 Visure

Visure[11] is one of the most impressive requirements management tools offering a comprehensive and complete SDLC platform including full trace-ability, integration with Microsoft Office applications, risk management, test management, bug tracking, requirements testing, requirements quality analysis, requirement versioning and base-lining, powerful reporting, and standard compliance templates.

These tools can be summarized to have following dominating specialties[12],

- **JamaSoftware** - Best for software development and testing.
- **Visure Requirements** - Best for configuration management.
- **Caliber**- Best for storyboards and simulations.
- **Pearls**- Best for team collaboration features.
- **Orcanos** -Best for visualization and reporting.

# 5 Phase Review

## 5.1 Asad Tariq

The Requirement Engineering phase took us more time than the first deliverable but I am very happy that we were able to finish it on time despite several of us had an exam during this week. Our communication remained the key, Although it was hard to find an appropriate meeting time because everyone was a bit occupied with their exams/quiz's but I am glad that everyone completed their tasks according to the decided deadlines. I'm looking forward to work on the code and I am sure if we keep on working this way we would be able to achieve more than the expectations.

## 5.2 Mahmoud Emara

I have a positive impression of this phase. Overall it wasn't that easy compared to the first task of the project initiation but we managed to come up with ideas for the requirements engineering but I believe that we managed the this phase quite well. We managed to set up a form of good communication on the first day of the sprint. We collectively looked at our strengths and weaknesses and decided the role selection on the collective input. We then went through all the required topics of the document and discussed each section on a broader level. Each section was then assigned to a particular member so that a more thorough version of our discussion would be written down. We then met a couple more times to present our sections to each other and suggest corrections/additions to the document. One aspect I think we still need to improve on is figuring out the optimal times for meeting each other online. Since all of our schedules are so diverse, it was quite difficult to find time where everyone could attend the meetings.'

## 5.3 Syed Faizan Hassan

The Requirements Engineering phase is a very important part as all our future work is dependent over it. It is important to look for each and every small detail and make sure that you do not miss out on any fundamental requirement. We divided the tasks into two group of people but had a collective meeting to review the requirements so that everyone's opinion,feedback, point of view can be considered.We also collected the feedback from the professor and now after finalizing the document I feel very confident that we are headed in the right direction. As the work in this phase was interdependent we had to set some tight deadlines in order to complete the phase within time.We smoothly completed the phase because everyone respected the deadlines and I am very happy with the performance of our team. I can't wait to start working on the code with such an amazing team.

## 5.4 Lukas Liß

Our complete team was well aware of the importance of the requirement engineering phase. Therefore we had multiple meetings to allow us to improve the requirements in an iterative manner. I really appreciated the openness with which we talked about our strength and weaknesses in order to come up with a good distribution of the tasks. As multiple team members had exams during this sprint we had to communicate clearly about this upfront. I think this worked well as we used Trello to keep track of the deadlines so everyone all ways knew when a task will be finished by whom. Additionally I have to mention here that everyone met their deadlines just like in the last sprint! I really appreciate this reliability. I myself had an exam last week and therefore I was really happy about the project structure we initiallised in the last sprint because it created a structure that allowed me plan ahead and manage the exam and the project tasks. Now that the requirements are planned I am very much looking forward to start coding and see our ideas turning into executable code.

### 5.5 Mina Khalid

The requirement engineering phase was an uphill task as compared to the previous deliverable but the entire team openly discussed their key strengths and we divided the tasks accordingly. Since this time everyone was super occupied with either their exams or with other deadlines, deciding meeting times was a bit challenging but we eventually worked around the solutions for this too. Everyone got done with their tasks in due time and we then as a team went through everything together to look for improvement areas in the compiled work. I really appreciate the prompt guidance of our professor Mr. Berti in sharing his feedback on the ambiguous points we raised with him. It feels like everyone is always on their toes to help the other person out. This creates a very healthy working environment for all of us. I am super excited and really looking forward to the next phase now and I am positive that our team with the leadership of Mr. Berti will deliver more than what is expected from us.

## References

[1] Wil van der Aalst. Data Science in Action. Springer Berlin Heidelberg, 2016.

[2] Leemans, S.J.J., Fahland, D. van der Aalst, W.M.P. Scalable process discovery and conformance checking. Softw Syst Model 17, 599–631 (2018). https://doi.org/10.1007/s10270-016-0545-x

[3] https://swtch.com/ rsc/regexp/regexp1.html

[4] Ulsan National Institute of Science and Technology. Process Mining at Seoul National University Bundang Hospital, 2014. https://www.tf-pm.org/resources/casestudy/process-mining-at-seoul-national-university-bundang-hospital.

[5] Reference for Word of Mouth Marketing Statistics: https://www.bigcommerce.com/blog/word-of-mouth-marketing/what-is-word-of-mouth-marketing

[6] https://www.orcanos.com/compliance/about-us

[7] https://www.jamasoftware.com/blog/four-fundamental-requirements-management-best-practices

[8] https://www.jamasoftware.com/platform/jama-connect

[9] https://pearls-inc.com/about-us

[10] https://www.businessprocessincubator.com/content/requirements-management-solution-caliber-micro-focus

[11] https://visuresolutions.com/alm-solution

[12] https://thedigitalprojectmanager.com/requirements-management-tools