Filterbank Project Documentation

Albert van Kiel, Timo Hermsen, Robin Baneke,
Max van Hasselt, Robert Kleef and Menno Prinzhorn

October 15, 2018

Chapter 1

Introduction

During the course of this project, we are creating a free and open source framework that contains the generic algorithms and file handling for astronomical data sets. This framework will be modular. Similar to OpenCV, wherein specific modules can be added and disabled depended on the needs of a project. This framework will be implemented in Python and C++.

1.1 Current situation

The eScience Center is in need of an application that is able to both read and write filterbank files. There already is an application in use. However, the current application is outdated and therefore has to be rewritten. The scope of this project is to rewrite a new software stack with updated technology, such as using the GPU which is highly modular and re-usable.

The development of the application is split into multiple parts and distributed over two different teams. Each team is assigned to work on one of these parts and make sure that it completed within the given timeframe. After each part is finished, it has to be merged into the main application.

Chapter 2

Plan of action

2.1 Mission

The purpose of this part of the project is to read and write filterbank files. We are first required to create this part using Python. And if we are finished with creating this part in Python, we will port it to C and C++.

We plan on using Scrum as our project method. By applying Scrum we are able to make changes, based on the feedback given by our product owners, whenever necessary to the application.

2.2 Monitoring performance

In order to monitor the performance of our development our product owners decided to plan weekly reviews. These reviews could be used to discuss the pace of our project. If our product owners are not satisfied with the pace of our development team, we could look at possible improvements.

2.3 Risks

- A shortage of knowledge about Python/C/C++
- An inflation of requirements
- A wrong estimation of required development time
- Lack of motivation

Chapter 3

Technical details

3.1 Requirements

The requirements can be divided into two different categories. There are both functional and non-functional requirements.

3.1.1 Non-functional

The non-functional requirements describe the technical requirements for the application.

- Support for:
  MacOS High Sierra (x86_64), CentOS (x86_64), Raspbian(ARM x64)
- Python 3.6
- Must be modular

3.1.2 Functional

- As user I want to read filterbank files in my program, so we can use the astronomical data in scientific programs.
- As user I want to write filterbank files in my program, so we can write astronomical data in scientific program
- As scientific programmer I want that my filterbank reader performs with very large datasets (+1TB of data), which is often the case in software
- As scientific programmer I want that my filterbank writer performs with very large datasets(+1TB of data), which is often the case in software