Radgraph - A tool for designing radar data processing schemes

Joonas Karjalainen[1] , Markus Peura[1] , Harri Hohti[1], Jarmo Koistinen[1]

[1]*Finnish Meteorological Institute, PO Box 503, 00101 Helsinki, Finnland*

(Dated: 25 April 2014)

1 Introduction

Radgraph is an application developed at The Finnish Meteorological institute (FMI) for designing radar data processing schemes. A complete radar processing scheme usually consist of multiple phases, where radar data is at first quality controlled and then processed to products that can be even further processed to higher level products. The implementation of different processing algorithms varies and in many cases the order, in which the algorithms are applied is not unambiguous. As a result, designing radar processing schemes can become very complicated task.

2 Features

Radar processing schemes contain information at different hierarchical levels. It is important that processing can be examined at high level to get an overview of what data is used at different processing steps and how the processes are linked together. It is also very important to be able to view the small scale details about the implementation of each sub process.

If the low and high level information is presented in one big graph, the overall view is most likely lost into the complexity of the it. On the other hand, if the sub processes are presented as separate graphs, valuable information about the connections between the processes is lost.

Radgraph offers a solution to this problem by allowing easy navigation from general overview to small scale details and back. Designing the processing schemes is made modular so that complicated tasks can be divided into smaller pieces that can be designed separately and then combined into a bigger graph.

Some sub processes might be absolutely necessary for obtaining correct end product while other steps might be just useful or optional. This is taken into consideration in the program so that the connections between the processing steps can be divided into different classes and they can be given different weight values depending on their importance. This information can then be used to highlight certain areas from the graph when focusing on some smaller part of the processing scheme.

In future releases, the program will contain a library of predesigned processing steps that can be freely combined and edited by the user. Processing graphs can also be saved to a format that can be easily distributed between the users.

3. Implementation

The program is written in python and the graphical user interface implemented using the kivy library. As a result, Radgraph can be run on multiple platforms Linux, Windows, OSX, Android and iOS. In addition to keyboard and mouse, Radgraph can also be controlled with touch screen.