Contents

Chapter 1

design

1.1 Plan

Below is the plan for storing the ENRAM workflow (i.e. the whole process from reading the radar data to analysis and visualization of the data):

- get all the C code and the IDL code, store it in a new repository 'enram' at `https://github.com`[1]
  - also store a selection of the data, e.g. one sample from every station, for unit testing

    Q: on what machine are the tests run? Whose license is IDL using then?
- set up a virtual machine ('ENRAMVM') containing a Linux operating system

  Q: what version of Linux? Ubuntu/Lubuntu? does it matter?
- copy the data to inside the virtual machine
- within the virtual machine, install IDL
  - get an idl license
- within the virtual machine, install hdfview, plus any other software that you need
- set the necessary paths
- check out the code from the enram repository using `git clone`

[1] We prefer a public project with a license like the ones we already have in place for existing projects. Check that this is OK with Martin, Hidde, Adriaan.

- save the virtual machine on the external hard drive
- also store the virtualbox installer on the external harddrive

1.2 End user steps to reproduce the ENRAM workflow

These are the steps that Willem needs to take in order to reproduce the ENRAM workflow.

- Plug in the external USB harddrive
- In Windows, open a file explorer, navigate to the USB drive, click on the virtualbox installer.
- In Windows, start virtualbox. Load/open the virtual machine 'ENRAMVM' located on the external haddrive.
- Log in if needed (can I skip logging in/maybe use a guest session or something?)
- Open a terminal
- `cd` to the directory that holds the local copy of the github/enram repository.
- In the terminal, start IDL
- At the IDL prompt, run `.r bird_call.pro` or any other program that you want.

1.3 Drawbacks

What can I not do with this setup? Are there any other drawbacks?

- external harddrive

  Q: (would it make sense to make a harddisk image that is then stored somewhere on, say, gridstorage?)

  A: yes

- unit testing/integration testing not allowed under the license...not possible?

Chapter 2

notes

package hdfview on linux ubuntu

- describe the METEOR 360AC (IEEE) C-Band setup/what the radar actually measures
- what the raw data look like
- what the converted data look like
- read the data
- what is a cluttermap
- how cluttermaps are calculated
- PPI = http://en.wikipedia.org/wiki/Plan_position_indicator
- http://en.wikipedia.org/wiki/C_band

Selex SI Gematronik rainbow

2.1 History

1. Adriaan's vol2bird C program
2. Martin's IDL code plus C-library

2.2 Overview of files and folders

When I plug in the external USB drive, these 12 directories currently exist at `/media/daisycutter/49f14219-4570-4c14-87b3-d7e502ed3736/users/graafdem/FLYSAFE2`:

1. bird/
2. cluttermaps/
3. knmi/
4. odim/
5. rainbow/
6. raw/
7. sens_clutter/
8. software/
9. statistics/
10. web/
11. zero_bird/

Here's a comparison between the contents of two dirs based on a recursive diff:

dir1='/home/daisycutter/enram/flysafe2-readonly/'

dir2='/media/daisycutter/49f14219-4570-4c14-87b3-d7e502ed3736/users/graafdem/FLYSAFE2/'

1. $dir1/idl/ and $dir2/software/idl are functionally equal. Differences are whitespace characters.
2. $dir1/process/ and $dir2/software/process
3. $dir1/visualisation/ and $dir2/software/visualisation: dir1 has a visualisation/profile/multi_prof_test.pro and a profile/test_gif

1. idl/
2. idl/clutter/
3. idl/io/
4. idl/vol2bird/
5. process/
6. process/log/
7. usr/people/graafdem/IDL/flysafe/*
8. visualisation/map/
9. visualisation/map/loop_gif/
10. visualisation/profile/

11. `visualisation/profile/all_gif/`
12. `visualisation/profile/test_gif/`

Below is a more detailed description of what each directory contains:

1. `idl`
   - contains 3 IDL scripts: 'common_definition.pro', 'radar_definitions.pro', and 'radar_names.pro'.
   - 'common_definitions.pro' defines a number of global (`COMMON`) constants like the maximum number of elevation scans, the radius of the Earth, a conversion factor to get from Z [1] to cm^2/km^3 at C-band [2] [3,4]
   - 'radar_definitions.pro' contains 5 hardcoded, absolute paths[5,6,7]:
     (a) `RAW_DATA_PATH` is the full path to the raw data, as delivered by the various radar operatives. Its current value is `/usr/people/graafdem/FLYSAFE/process/data/raw/`, which does not exist on my system or the virtual machine.
     (b) `IO_PATH` is the full path to the IDL io definitions. Its current value is `/usr/people/graafdem/FLYSAFE/idl/io/`, which does not exist on my system or the virtual machine.
     (c) `INPUT_DATA_PATH` is the full path to the harmonized, ODIM format data. Its current value is `/usr/people/graafdem/FLYSAFE/process/data/odim/`, which does not exist on my system or the virtual machine.
     (d) `CLUTTER_DATA_PATH` is the full path to the cluttermap files. Its current value is `/usr/people/graafdem/FLYSAFE/process/data/cluttermaps/`, which does not exist on my system or the virtual machine.
     (e) `BIRD_DATA_PATH` is the full path to the output data directory. Its current value is `/usr/people/graafdem/FLYSAFE/process/data/bird/`, which does not exist on my system or the virtual machine.

     In brief this is how I think the components from these directories work together:
     (a) The starting point is of course the data in `RAW_DATA_PATH`;
     (b) The data are stored in various formats and styles, so in order to access them, one needs to know how to read each specific format and style, which

[1] reflectivity power?
[2] there are a couple of integer variables that have an L after their value, e.g. `NLAYER=30L;`, according to http://www.exelisvis.com/docs/IDL_Data_Types.html this means long integer in IDL. Maybe the programmer adopted a 'better safe than sorry' approach? (Normal integers are 16 bit, i.e. -32768 to +32767).
[3] I should check 'common_definitions.pro' again sometime when I have a better idea of what each variable is.
[4] Also, the COMMON block in 'idl/vol2bird/bird_call.pro' is not exactly equal to that found in 'idl/common_definition.pro'. Is it necessary to have both versions?
[5] maybe check out http://www.physics.wisc.edu/~craigm/idl/archive/msg06372.html
[6] does IDL need hardcoded paths or could it use relative paths?
[7] am I free to re-organize the directory structure?

is defined in the algorithms in `IO_PATH`. After the data have been 'harmonized', i.e. converted to a uniform format (ODIM/HDF5), they are saved in `INPUT_DATA_PATH`;

(c) Clutter (unwanted return signals from buildings, vegetation, as well as foreign signals that fall within the radar's receiver wavelength) needs to be removed from the data with the use of so called cluttermaps (i.e. masks). Before that can be done, the cluttermaps need to be calculated, essentially by picking a time interval during which all return signals can be attributed to clutter. The resulting maps are stored in directory `CLUTTER_DATA_PATH`;

(d) By using the cluttermaps from `CLUTTER_DATA_PATH` and the harmonized data from `INPUT_DATA_PATH` <some algorithm[1]> is used to calculate how many birds are airborne in the vicinity of each radar.

- 'radar_names.pro' contains a function that returns and optionally prints an alphabetical list of radar station codes when given either the full name of a country, the country code, a list of radar station codes, or an arbitrary combination.

2. `idl/clutter`

   contains

3. `idl/io`

   contains the input/output routines for the different radar systems are defined, per country (i.e. it is assumed that all systems from one country can be handled equally).

4. `idl/vol2bird`

   contains

5. `process`

   contains

6. `process/log`

   contains

7. `usr/people/graafdem/IDL/flysafe/`

   contains

8. `visualisation/map`

   contains

9. `visualisation/map/loop_gif`

   contains

[1] probably `idl/vol2bird`?

10. `visualisation/profile`

    contains

11. `visualisation/profile/all_gif`

    contains

12. `visualisation/profile/test_gif`

    contains

2.3 Proposed directory structure

1. data
2. data/raw
3. data/harmonized
4. data/harmonized/cluttermaps
5. data/harmonized/odim
6. data/harmonized/bird
7. src
8. src/clutter
9. src/io
10. src/vol2bird

Appendices

Bibliography

Index