OPEN SOFTWARE: INITIAL EXPERIENCES WITH THE INTEGRATION OF THE BALTRAD TOOLBOX AND LROSE AS A SERVICE

AMOS National Radar Workshop

13-14 Nov 2019, Clayton, VIC, Australia
OPEN AND COMMERCIAL SOFTWARE

• How does co-existence of Open SW and commercial SW affect the SW development?

• How can Open SW be integrated into commercial SW?
  ➢ Data processing and display
  ➢ Relevant processing steps
  ➢Legal issues (exe vs. lib)

• How about maintainability and availability for years to decades?
  ➢ Training: Learn and understand
  ➢ Bug tracking and fixing
  ➢ OS and system environment
PROPRIETARY SW: LEONARDO RAINBOW5

• Client-Server architecture
• C++ with support-libs
  ➢ Linux
  ➢ Windows
• Proprietary data format
  ➢ XML based
  ➢ Data files with BLOBs
• Version 5.x.y
  ➢ First (major) release 2004
  ➢ 51 more (major) releases since
• 2019: Almost no open SW in data processing and display
  ➢ Let’s get started
BALTRAD TOOLBOX - HTTP://GIT.BALTRAD.EU/
BALTRAD "Toolbox"

- No interaction with radar
- No own visualization
BALTRAD TOOLBOX

• **Python** Environment → Python version issues

• Can be used **offline** (without connection to a BALTRAD node), interactively through a python interpreter as well as running binaries on the command line

• Packages for data preprocessing and product generation
  • **bRopo**: Identification and removal of non-precipitation echoes in polar data
  • **Beamb**: Determination of beam blockage and correction
  • **RAVE**: Identification and removal of non-precipitation echoes and attenuation correction in polar data

• **ODIM HDF-5** file I/O
TEST BALRAD TOOLBOX WITH RAINBOW5

1. **Convert** Rainbow raw data to ODIM_H5 format

2. **Process** ODIM_H5 data with BALTRAD Toolbox

3. **Convert** ODIM_H5 (either raw data or Cartesian product) to Rainbow format

4. **Display** converted products or generate products based on converted raw data data with RainDART
# SUMMARY (NEWCOMER’S VIEW)

## BALTRAD Toolbox

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☺️</td>
<td>Possible to combine with Rainbow</td>
</tr>
<tr>
<td>☻️</td>
<td>Several preprocessing products useful</td>
</tr>
</tbody>
</table>
| 😞 | Documentation missing/bad for bropo package  
Parameters for algorithms sometimes not clear  
Difficult to find the right parameter combination for a good performance  
(default values sometimes show bad results) |
| 😞 | Some difficulties with the installation, e.g. conflict of different python versions |
## SUMMARY (NEWCOMER’S VIEW)

<table>
<thead>
<tr>
<th>Package</th>
<th>😊</th>
<th>😞</th>
<th>😊 / ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>bRopo</td>
<td>bRopo-sun</td>
<td>bRopo-clutter</td>
<td>bRopo-speck</td>
</tr>
<tr>
<td></td>
<td>bRopo-emitter2</td>
<td>bRopo-clutter2</td>
<td>bRopo-SpeckNormOld</td>
</tr>
<tr>
<td></td>
<td>bRopo-ship</td>
<td>bRopo-emitter</td>
<td>bRopo-softcut</td>
</tr>
<tr>
<td></td>
<td>bRopo-lookup</td>
<td></td>
<td>bRopo-biomet</td>
</tr>
<tr>
<td>beamb</td>
<td>beamb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAVE</td>
<td>radvol-att</td>
<td>radvol-speck</td>
<td>radvol-spike</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>radvol-broad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>dealias</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hac-filter</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>radarcomp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>hmc</td>
</tr>
</tbody>
</table>
EXAMPLE: WHAT ALGORITHM SHALL I TAKE?

Interference and sun removal:

- bRopo:
  - Emitter
  - Emitter2
  - Sun
  - Sun2

- RAVE-RADVOL:
  - Spike
EXAMPLE: MISSING MAINTENANCE

BALTRAD HMC

- Fuzzy logic hydrometeor classification from dual pol raw data
- Not (yet?) included in the available BALTRAD package
- BALTRAD lab link http://git.baltrad.eu/lab with suggested test environment does not exist
AND EXAMPLE: BROPO INSECT REMOVAL

Parameters:
• dbz_max
• dbz_delta
• alt_max in meters
• alt_delta in meters

Example:
Bottom left: bRopo biomet=10,10,5000,1
Bottom right: bRopo biomet=20,10,5000,1
EXAMPLE: RAVE DE-ALIASING
LROSE OVERVIEW

• 4-year joint project between the Atmospheric Science Department at CSU and the Earth Observing Laboratory (EOL) at NCAR, funded by the US National Science Foundation

• Open source software to the community of scientists, researchers and operational organizations using atmospheric lidars, radars and profilers. The code is freely available on the web under a BSD-style license

• Organized into modular libraries and applications, mainly C++

• Includes TITAN

• LROSE is intended to handle information at all stages:
  ➢ raw time series data at the instrument in native format
  ➢ moments data in radial coordinates
  ➢ algorithms in radial coordinates
  ➢ products in Cartesian coordinates
  ➢ engineering displays
  ➢ science displays that integrate with other data sets for visualization
  ➢ provision of QC data to models for data assimilation
TEST LROSE WITH RAINBOW5

1. Convert Rainbow raw data to CfRadial format (e.g. with the LROSE application RadxConvert)

2. Process CfRadial data with LROSE applications
   - Some use different formats, e.g. TITAN. Use LROSE converters from/to CfRadial

3. Convert CfRadial to Rainbow format (with Rainbow converter → currently only for raw data, not for products)
   - Rainbow converter currently only for raw data, not for products
   - This presentation shows LROSE display

4. Display generated products (based on converted raw data) with RainDART
## SUMMARY (NEWCOMER’S VIEW)

### LROSE Toolbox

<table>
<thead>
<tr>
<th>Icon</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>😊</td>
<td>Quite sophisticated tool with useful products (precipitation estimation, attenuation correction, Echo classification, Storm tracking and analysis,…)</td>
</tr>
<tr>
<td>😊</td>
<td>Modular structure, tools can be executed individually</td>
</tr>
<tr>
<td>😞</td>
<td>Handling with parameter files is quite inconvenient at least for visualization tools, as parameters/input have to be adjusted each time</td>
</tr>
<tr>
<td>😞</td>
<td>Different data formats (CfRadial, MDV) for different applications (Radx, Titan), conversion necessary</td>
</tr>
<tr>
<td>😞</td>
<td>Documentation is good, but multiple sources and many parameters for each application are shortly described only in the parameter file</td>
</tr>
<tr>
<td>😞</td>
<td>Conversion of CfRadial products to Rainbow to be done</td>
</tr>
</tbody>
</table>
SUMMARY ON TESTED PACKAGES (NEWCOMER’S VIEW)

- **RadxBeamBlock**

- **RadxRate** → KDP calculation, Z and ZDR attenuation correction, Particle Identification (PID), Precip Rate calculation

- **Radx2Grid** → coordinate transformations for ground-based radars from a spherical grid on which radar data is collected

- **RadxKdp** → KDP calculation, Z and ZDR attenuation correction

- **RadxPid** → KDP calculation, Z and ZDR attenuation correction, Particle Identification (PID)

- **RadxQpe** → KDP calculation, Particle Identification (PID), beam blockage calculation, Precipitation

- **Hawk-Eye** → polar raw data Display

- **TITAN tracking** → Rview Display, Storm header and storm data files
EXAMPLE: RADXRATE
EXAMPLE: RADXRATE: ATTENUATION CORRECTION
EXAMPLE: RADXRATE-PID @HAWK-EYE
Precip rate at surface
EXAMPLE: RADXBEAMBLOCK

Max. terrain height per gate (PEAK)

Cumulative beam extinction (BEAME)
**SUMMARY: FIRST TESTS WITH BALTRAD AND LROSE**

- Open Source Software offers algorithms and tools that do not exist in Rainbow5
- Several algorithms exist in both Open SW and RB5
- Configuration of Open SW is somewhat painful
  - No big deal for the experienced user
- Both BALTRAD and LROSE algorithms can be integrated into RB5
  - Converters need to be extended
  - Rainbow Display would need to be extended for some LROSE and TITAN output (e.g. LROSE PID, TITAN-Storm-Tracks)
- In short: Combination of RB5 and Open SW is very promising
Thank you!

Ronald Hannesen

R.Hannesen@selex-es-gmbh.com