Karlsruhe Institute of Technology

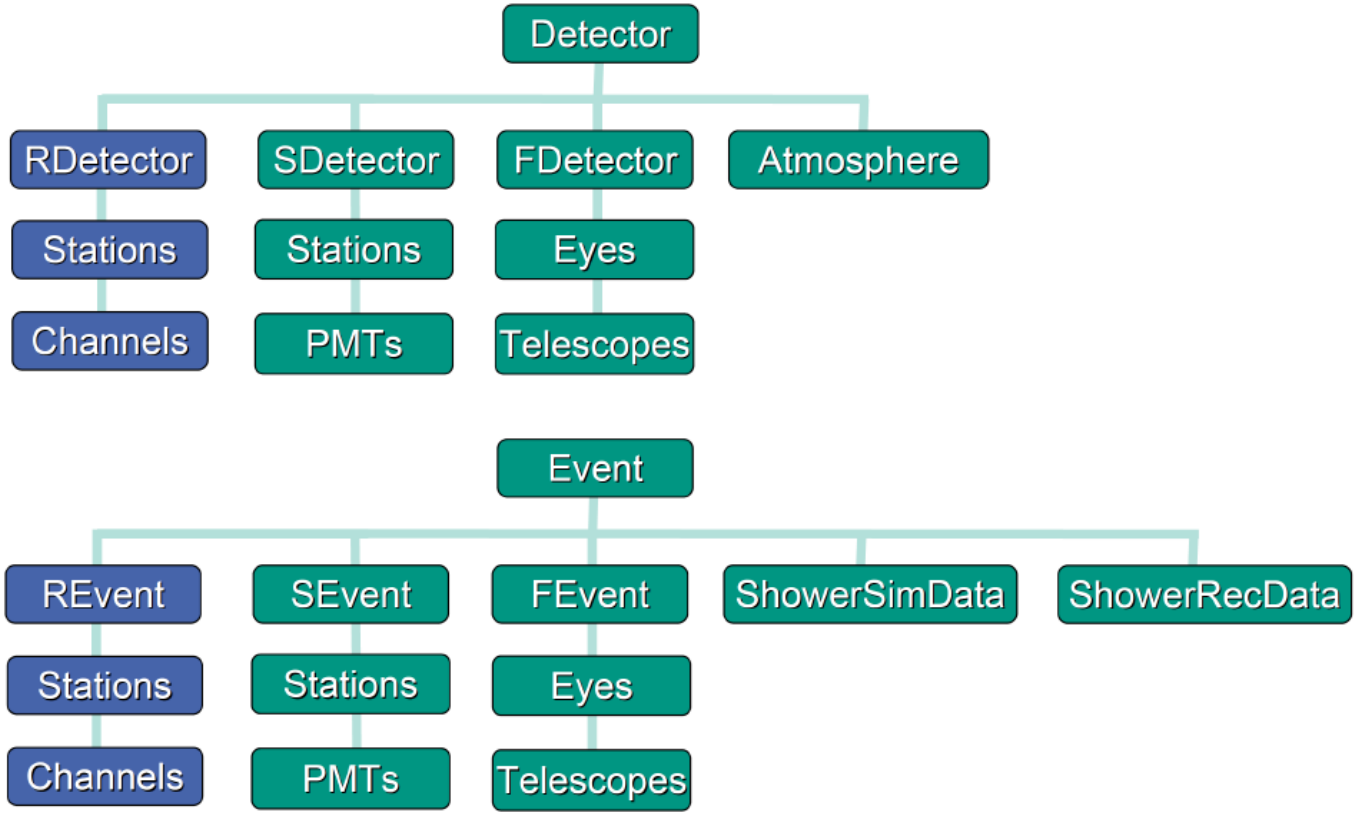
Radio functionality in Offline and issues re the upgrade

T. Huege

Radio in Offline right now

- read-in of various data formats
  - binary AERA data (no longer maintained)
  - aerarootio data (e.g. for periodically triggered)
  - ioauger data, especially merged xrad files
  - various simulation data formats, only CoREAS actively maintained
- event and detector classes alongside baseline detectors
  - difference: reconstruction quantities in ParameterStorage objects

Radio functionality in Offline



- seamlessly incorporated radio functionality next to SD and FD
- retained the logical hierarchy of the radio detectors

Principle of radio analysis with Offline

- example analysis pipeline for experimental data and simulations

Experimental Data

1. read file (AERA, …)

Simulations

1. read file (CoREAS, …)
2. fold in antenna response
3. add measured radio noise
4. fold in electronics response
5. resample to given time-base

6. correct for electronics response
7. suppress narrow-band noise
8. up-sample data (physical interpolation)
9. reconstruct physical e-field vector using correct antenna characteristics
10. post-process data, e.g. envelope

Radio in Offline right now

- event reconstruction
  - radio-only with iterative direction fit (discontinued, problems with RFI)
  - hybrid reconstruction using SD geometry early on
    - RdObserver and RdSimulationObserver for θ<=60°
    - RdHASObserver and RdHASSimulationObserver for θ>60°
      but: missing HAS-optimized radio reconstruction, work in progress

The RdObserver standard reconstruction

`<module> EventFileReaderOG </module>`	read in the data file
`<module> RdEventPreSelector </module>`	preselect data (externally triggered, min 3 stations)
`<module> EventCheckerOG </module>`	SD reconstruction
`<module> SdPMTQualityCheckerKG </module>`	
`<module> TriggerTimeCorrection </module>`	
`<module> SdCalibratorOG </module>`	
`<module> SdBadStationRejectorKG </module>`	
`<module> SdSignalRecoveryKLT </module>`	
`<module> SdEventSelectorOG </module>`	
`<module> SdPlaneFitOG </module>`	
`<module> LDFFinderKG </module>`	
`<try>`	
`<module> SdHorizontalReconstruction </module>`	
`</try>`	
`<module> RdEventInitializer </module>`	initialize radio event (coordinate systems, time windows, …)
`<module> RdStationPositionCorrection </module>`	attempt to fix data taken with wrong GPS settings
`<module> RdStationRejector </module>`	reject detector stations with known problems
`<module> RdChannelADCToVoltageConverter </module>`	convert ADC values to Voltage at ADC
`<module> RdChannelSelector </module>`	select channels to be used (not really used, but hi/lo-gain)
`<module> RdChannelPedestalRemover </module>`	remove DC offset (frequency 0 Hz)
`<module> RdChannelResponseIncorporator </module>`	deconvolve cables, filters, amplifiers -> V. at antenna footpoint
`<module> RdChannelBeaconTimingCalibrator </module>`	correct GPS clock drifts using beacon reference transmitter
`<module> RdChannelBeaconSuppressor </module>`	suppress beacon narrow-band lines from frequency spectrum
`<module> RdStationTimingCalibrator </module>`	correct remaining timing offsets (e.g. LPDA vs. butterfly ant.)
`<module> RdStationTimeWindowConsolidator </module>`	consolidate signal- and noise windows after timing shifts
`<module> RdChannelTimeSeriesTaperer </module>`	suppress artifacts by windowing the edges of the time-trace
`<module> RdChannelBandstopFilter </module>`	suppress narrow-band RFI (fixed known freqs. and automatic)
`<module> RdChannelUpsampler </module>`	recreate fine-binned time-trace from Nyquist-sampled data
`<module> RdChannelRiseTimeCalculator </module>`	calculate the signal rise-time in each antenna channel
	processing on Channel level done

```
<module> RdAntennaChannelToStationConverter </module>   reconstruct traces of 3d E-field vector – uses SD-reconstructed direction
<module> RdStationSignalReconstructor       </module>   search and quantify radio pulses in tje E-field trace of each station
<module> RdStationEFieldVectorCalculator    </module>   calculate mean orientation of E-field wrt. expectation

<loop numTimes="unbounded">
  <module> RdTopDownStationSelector </module>   ┐ iteratively deselect stations and retry plane wavefront fit to radio pulses
  <module> RdPlaneFit               </module>   ┘ to find and flag outliers caused by background pulses
</loop>

<module> RdClusterFinder                  </module>   deselect isolated stations with pulses due to background signals
<module> RdPlaneFit                       </module>   reconstruct arrival direction using a plane wavefront

<module> RdStationRiseTimeCalculator      </module>   calculate the signal rise-time of the electric field vector

<module> RdEventPostSelector              </module>   deselect events without successful direction reconstruction
<module> RdLDFMultiFitter                 </module>   fit two types of one-dimensional radio LDFs
<module> Rd2dLDFFitter                    </module>   fit two-dimensional radio LDF, basis of energy and Xmax reconstruction

<try>
  <module> MdMuonCounterAG          </module>   ┐
  <module> MdModuleRejectorAG       </module>   │
  <module> MdEventSelectorAG        </module>   ├ AMIGA muon reconstruction
  <module> MdLDFFinderAG            </module>   ┘
</try>

<try>
  <module> FdCalibratorOG                 </module>   ┐
  <module> FdEyeMergerKG                  </module>   │
  <module> FdPulseFinderOG                </module>   │
  <module> FdSDPFinderOG                  </module>   │
  <module> FdAxisFinderOG                 </module>   ├ FD reconstruction
  <module> HybridGeometryFinderOG         </module>   │
  <module> HybridGeometryFinderWG         </module>   │
  <module> FdApertureLightKG              </module>   │
  <module> FdEnergyDepositFinderKG        </module>   │
  <module> FdProfileReconstructorKG       </module>   ┘
</try>

<module> RdStationTimeSeriesWindowCutter  </module>   only retain small time-window around found pulse (lower ADST size)
<module> RdStationTimeSeriesTaperer       </module>   apply window to reduce artifacts from cutting trace
<module> RdREASSimPreparator              </module>   create input files to simulate this event with CoREAS
<module> EventFileExporterOG              </module>   extract this event in raw (ioAuger-xrad) file format
<module> RecDataWriterNG                  </module>   write reconstruction quantities into ADST file
```

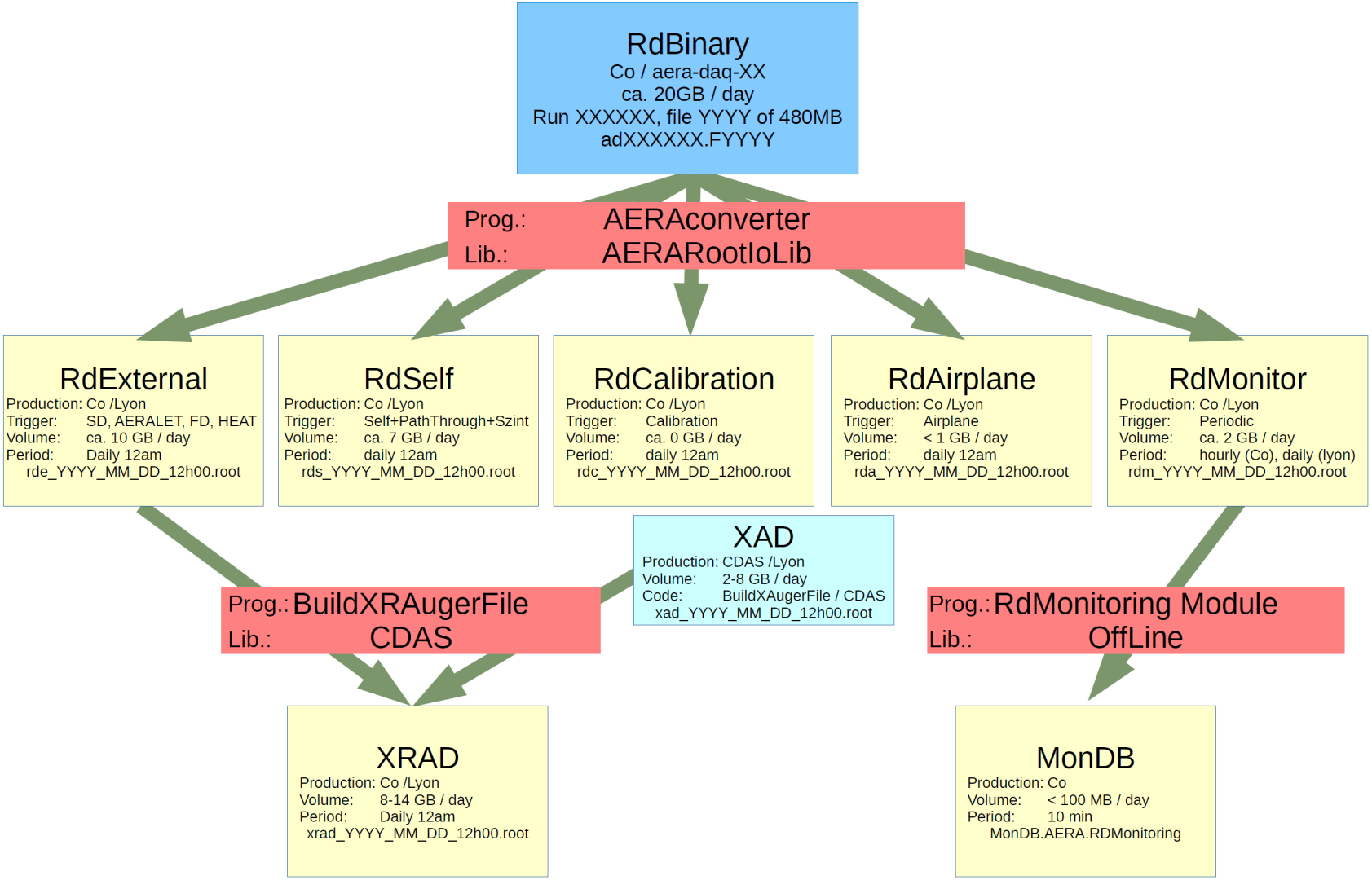
Radio in Offline right now

- validation tests RdCheck and RdSimCheck based on Rd(Sim)Observer
  - caveat: dependent on SD, i.e. fail when changes appear in SD
- our (the Collaboration‘s) track record of keeping the buildbot happy is not good

The Radio Upgrade and Offline

- the existing analysis functionality will work 1:1
- we currently only know of two issues
  - the logic of the RD data stream will change
  - the 1600 radio detectors will need to be defined

Data stream logic for AERA



Data stream logic for the Radio Upgrade

- full integration of radio data in CDAS data stream
- needs extensions similar to inclusion of SSD data
- need to think about the handling of calibration/background data
  - cannot use same logic as in AERA (periodic events)
- Julian will give more details in the next talk

Considerations on Radio Detector description

- in the early days we used XML files to define
  - positions of RD stations
  - antenna types and orientation, including gain pattern
  - response of analogue chain (filters, amplifiers, …)
  - ADC characteristics
- in A*E*RA hardware was often changed, detector is time-dependent
  - MySQL database with commissioning and decommissioning of components
  - increased complexity, not easily human-readable, higher hurdle to modify
- for the Radio Upgrade, both of these solutions can be re-used
  - XML files if no time-evolution, in particular if no need to describe detectors individually because all are „the same“
  - MySQL database if individual detector configuration, can then keep track of component replacements, e.g. if LNA breaks
  - probably need to wait and see how uniform detectors are
- disadvantage: logic of „integrated detector“ not represented in Offline, but I personally think this is not necessary and saves a lot of work

Summary

- almost everything can be re-used
- we will have to deal with RD data in CDAS stream
- existing solution can be used for Rd Upgrade Detector Description

- aside: we have to do better with keeping the buildbots happy