RadioOffline

ARENA Workshop
Rome, 25th June 2008

Julian Rautenberg
Bergische Universität Wuppertal
for the Auger Collaboration

Auger Radio

- Auger established a Radio Detection R&D Task Force
- Beginning Sep. 2006 data acquisition (DAQ) started using test-setups
- 3 different DAQ system used with varying set-ups (reported in this conference):
  - Techniques and results of the LOPES-STAR experiment for detection of UHECR by their radio emission
    Hartmut Gemmeke
  - Radiodetection of cosmic air showers with an autonomous radio detector installed at the Pierre Auger Observatory.
    Benoit REVENU
  - Observation of Radio Signals from Air Showers at the Pierre Auger Observatory
    Jose Coppens

Auger Radio Data

- Data exists now for three different data formats
- Stored centrally
- Setup debugging phase
- EAS have been measured *--- they are in the data!*

reconstruction of EAS-events

Within the main task of establishing radio-technique at Auger:

- Data comparison of different set-ups
- EAS measurement comparison with Auger

Radio data analysis

- Setup debugging phase:

  mainly stand-alone analysis of data

  e.g. determine background, test trigger strategies

- EAS measurement phase:

  reconstruction of events

  i.e. reconstruction of EAS quantities
  like arrival direction

... how is it done in Auger?

Combined Radio & Auger analysis: why?

Why integrate Radio into the Auger analysis framework?

- easy comparison with Auger measurement (SD/FD)
- unique event handling / combined Radio, SD & FD event
- common framework for the different test-setups
- easy comparison between test-setups
- ...
- hybrid / superhybrid event reconstruction

Combined Radio & Auger analysis: how?

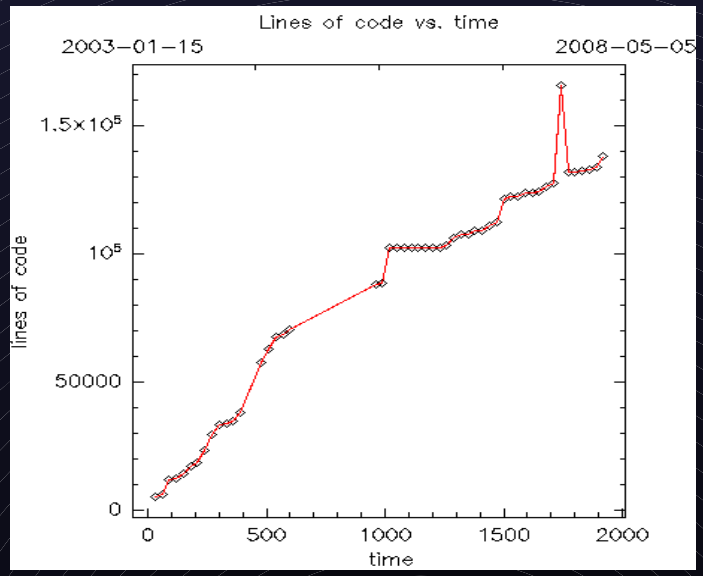
- examinating the structure of the radio-detector
- radio shows many similarities to surface detector (SD):

Surface detector	Radio
Stations (tanks)	Antenna Stations (poles)
3 PMT's	2 Antenna Channels (N-S, E-W)
time-trace of data	time-trace of data

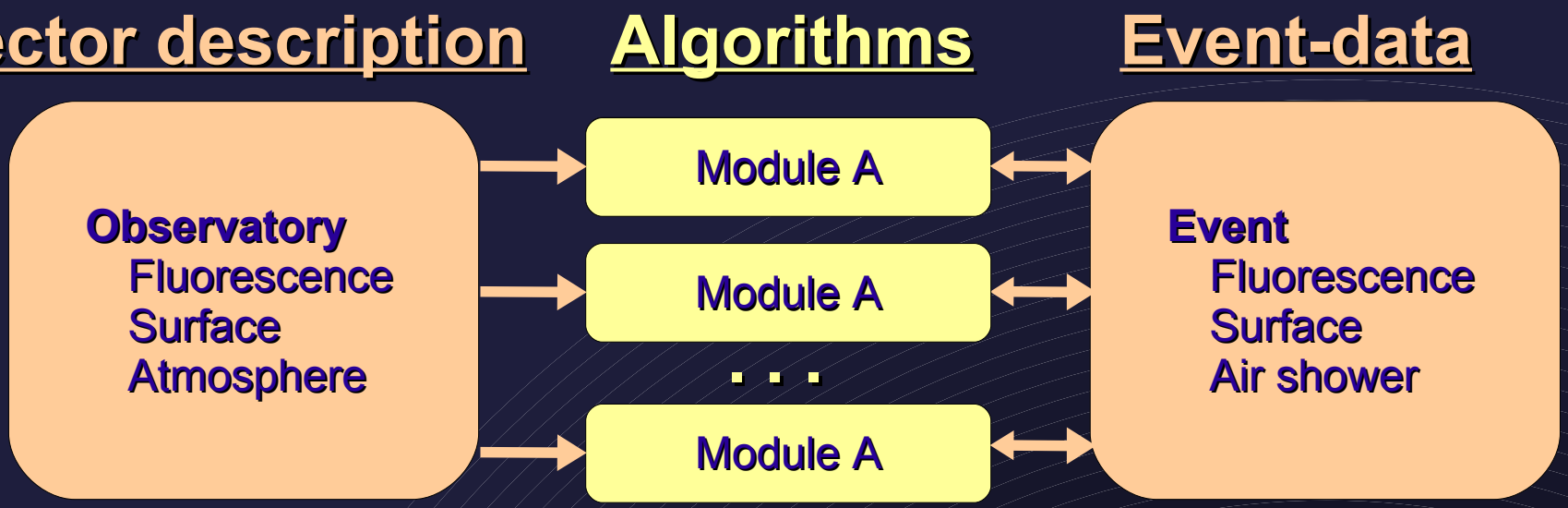
- SD implementation uses Auger utility-classes and lot's of interfaces
- use right framework right away to avoid stand-alone solutions

Auger Offline

- Main framework developed for analysis of Auger data
- Nucl. Instrum. Meth. A 580, 1485-1496 (2007)
- Started Jan 2002
- Since then: 1971 days
  - actual revision: 7355
  - lines of code: 150k
- C++, object oriented design and common open source tools used

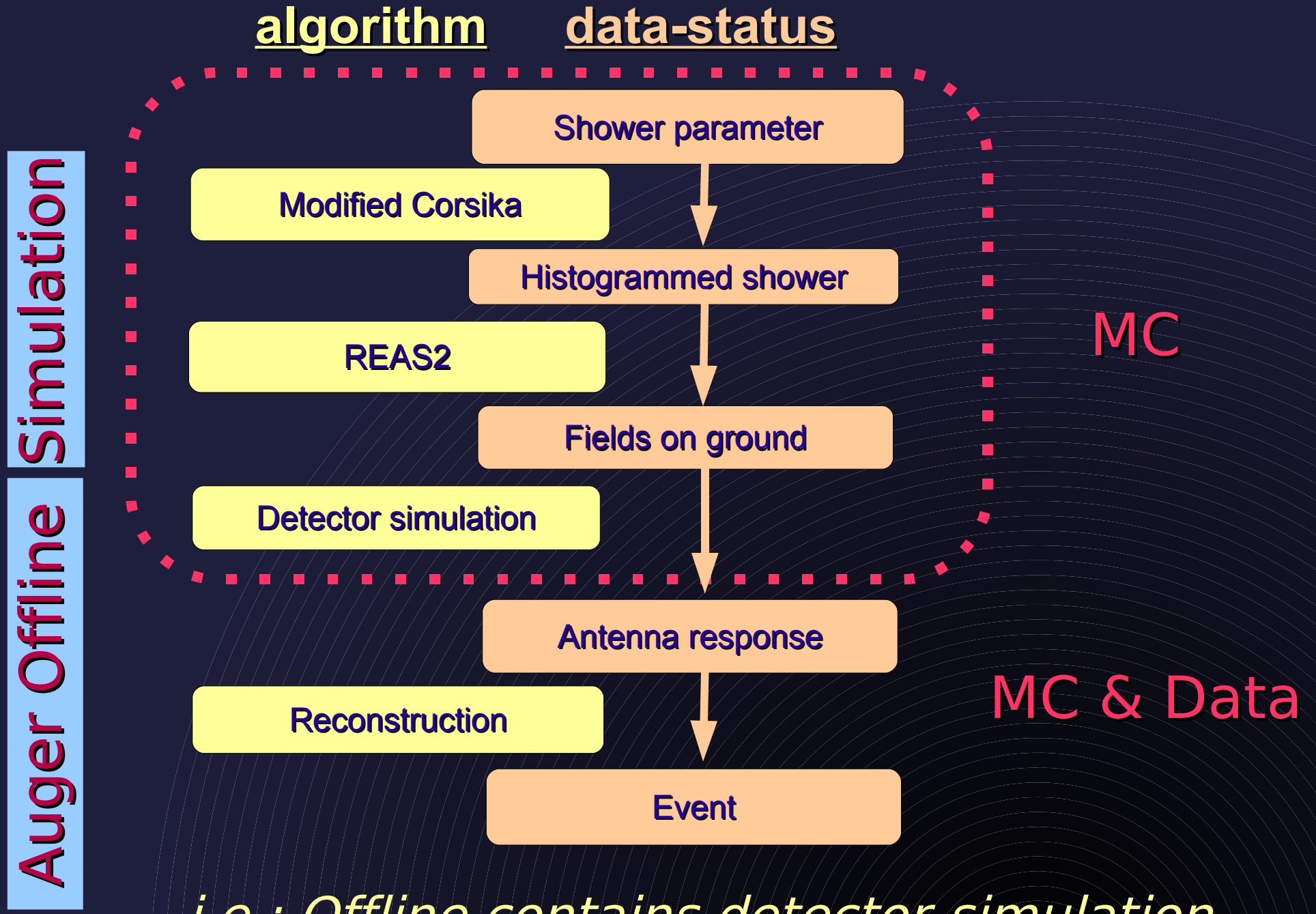


Auger Offline: Main framework components



- Modules:
  - Each step encapsulated in a module
  - Modules pass data through the event
  - Many applications by reusing modules
- Event:
  - Modules read from / write to the Event
  - Persistency-machinery
- Detector Description:
  - detector configuration and performance as function of time

Data-flow in Offline for radio simulation



i.e.: Offline contains detector-simulation

Offline modularity for Radio simulation:

- Modules:
  - RadioEventReaderA ↔ RadioEventReaderB
  - RadioSimulatorA ↔ RadioSimulatorB
  - RadioNoiseSimulatorA ↔ RadioNoiseSimulatorB
  - AntennaSimulatorA ↔ AntennaSimulatorB
  - RadioNoiseFilterA ↔ RadioNoiseFilterB
  - RadioReconstruction
  - ...

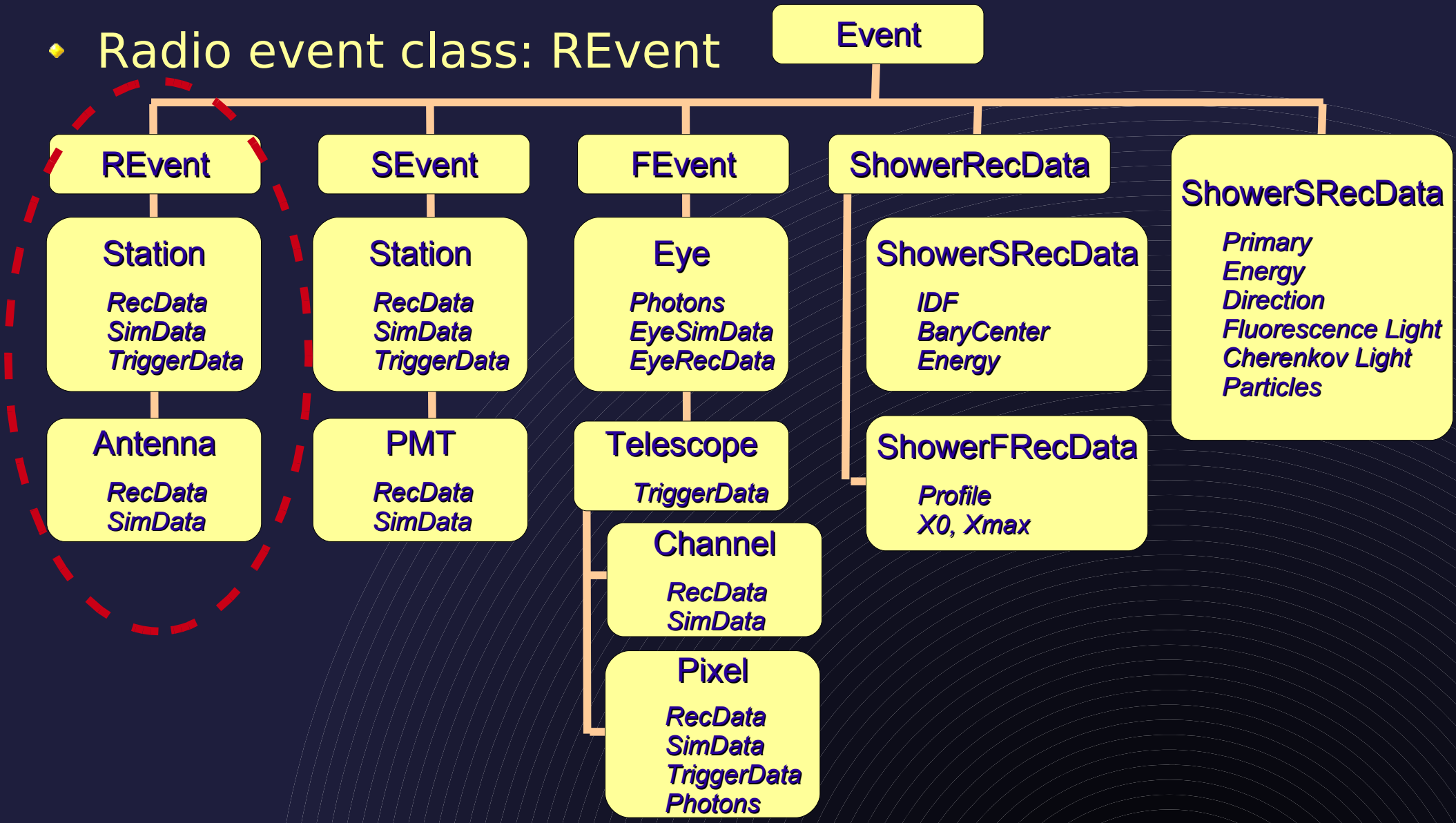
easy to exchange

... easy comparison between different

- *set-ups*
- *simulation models*
- *detector effects*

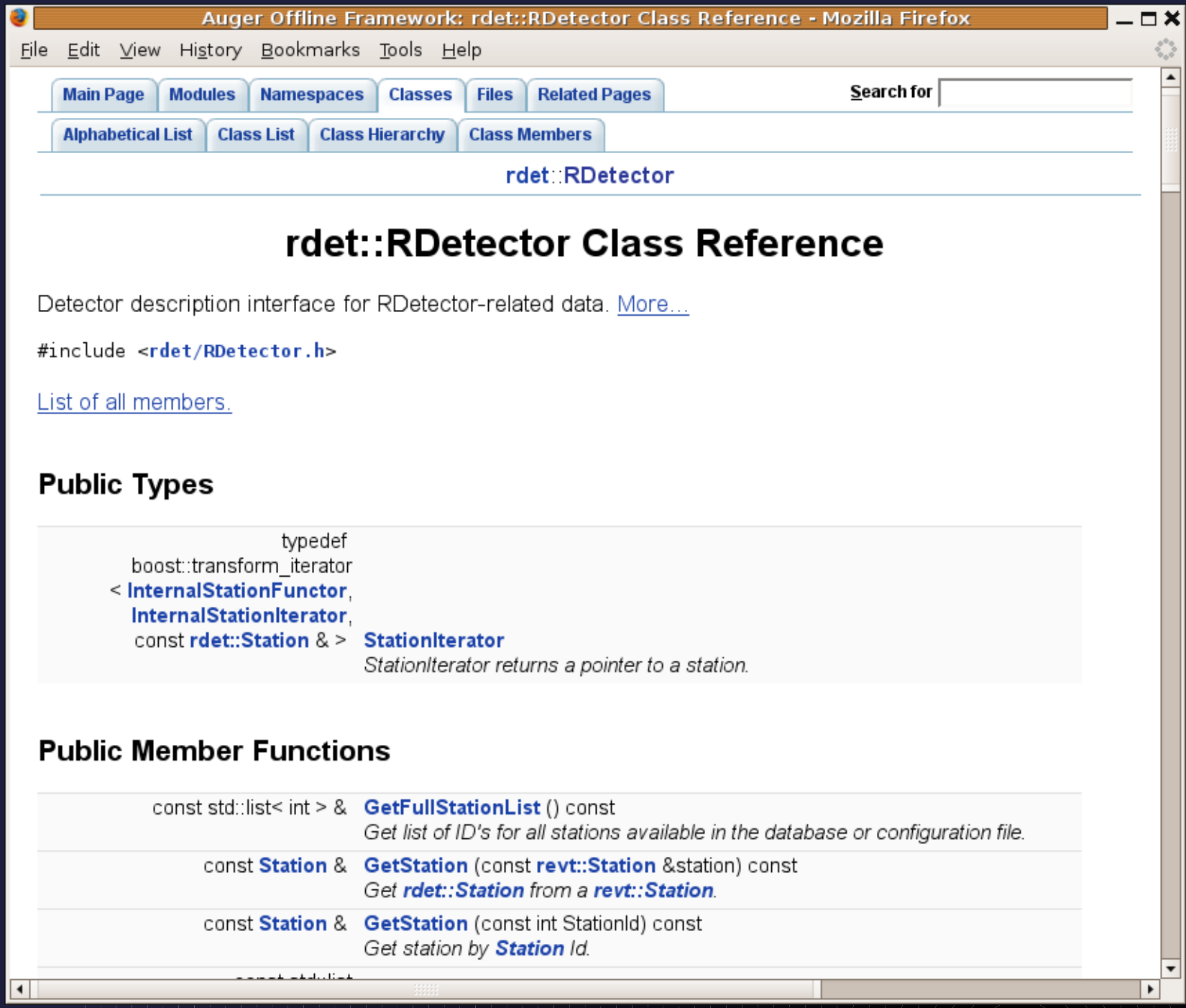
Radio additions:

- Radio event class: REvent



- Offline Event-Detector separation: RDetector
- Name-spaces: revt, rdet

RDetector


Auger Offline Framework: rdet::RDetector Class Reference - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Main Page Modules Namespaces Classes Files Related Pages
Search for
Alphabetical List Class List Class Hierarchy Class Members
rdet::RDetector
rdet::RDetector Class Reference
Detector description interface for RDetector-related data. More...
#include <rdet/RDetector.h>
List of all members.
Public Types
typedef boost::transform_iterator < InternalStationFunctor, InternalStationIterator, const rdet::Station & > StationIterator
StationIterator returns a pointer to a station.
Public Member Functions
const std::list< int > & GetFullStationList () const
Get list of ID's for all stations available in the database or configuration file.
const Station & GetStation (const revt::Station &station) const
Get rdet::Station from a revt::Station.
const Station & GetStation (const int StationId) const
Get station by Station Id.

revt::Station

Auger Offline Framework: revt::Station Class Reference - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Main Page | Modules | Namespaces | Classes | Files | Related Pages | Search for

Alphabetical List | Class List | Class Hierarchy | Class Members

revt::Station

revt::Station Class Reference

class to hold data at Station level More...

```
#include <revt/Station.h>
```

List of all members.

Public Types

typedef boost::indirect_iterator < InternalAntennaIterator, Antenna & >	**AntennaIterator** *Iterator over station for read/write.*
typedef boost::indirect_iterator < InternalConstAntennaIterator, const Antenna & >	**ConstAntennaIterator** *Iterator over station for read.*
enum	**SignalComponent** { **eTotal** = 0, **eElectron**, **ePhoton**, **eMuon**, **eHadron**, **eDecayElectron**, **eDeltaRay**, **eBackgroundElectron**, **eBackgroundPhoton**, **eBackgroundMuon**, **eBackgroundHadron**, **eAntennaGlassLight**, **eAlbedo**, **eDirectLight**, **eFirstBounce**, **eSecondBounce**, **eThirdBounce**, **eEstimatedUpperBound**, **eEstimatedLowerBound**,

RadioOffline: Status

- RadioOffline general structure implemented in Offline
- optional extension of standard Offline
- REvent and RDetector in analogy to SD introduced

Plans:

- finalize main classes
- implementation of modules for different data/simulation inputs, etc.
- utilize stand-alone code developed so far

... use it for Analysis

Summary

Radio Auger in EAS-measurement stage

reconstruction of Radio signals within Auger
for combined (hybrid) measurement
and easy comparison

implementation of Radio in
general Auger Offline framework

... that is RadioOffline