



AREUS

A Software Framework for ATLAS Readout Electronics Upgrade Simulation

Nico Madysa, on behalf of the ATLAS Liquid Argon
Calorimeter Group

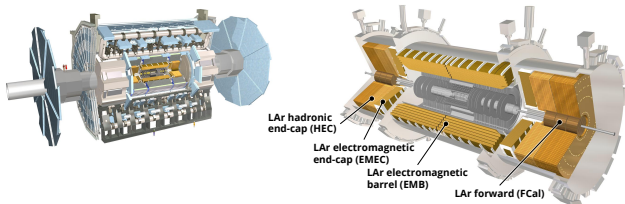


Federal Ministry of
Education
and Research



CHEP 2018, Sofia, Bulgaria, July 9–13

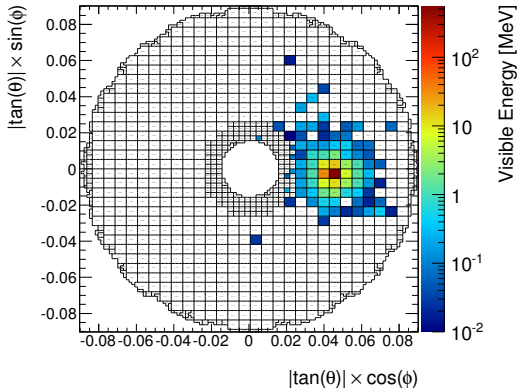
Upgrade of the ATLAS LAr Calorimeters Readout



- LHC will upgrade to *High-Luminosity LHC*
 - luminosity: $1 \times 10^{34} \rightarrow 7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$
 - pp interactions: 20 \rightarrow 200 per bunch crossing
 - trigger rate: 100 kHz \rightarrow 1 MHz
- full LAr calorimeter readout will be replaced in 2024–2026
- detailed simulation of new readout necessary:
 - minimization of noise
 - suppression of pile-up effects

\Rightarrow *AREUS Framework*

AREUS Input: Calorimeter Energy Deposits

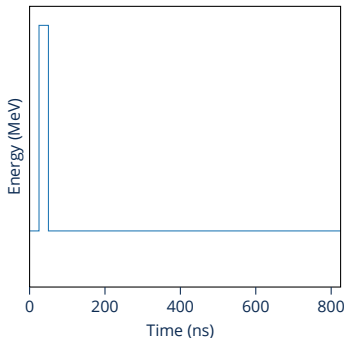


- from detailed Geant4 simulation of the detector
- event $\hat{=}$ list of detector cell hits
- each cell simulated separately
- pile-up: multiple events per bunch crossing

The Readout Chain in AREUS



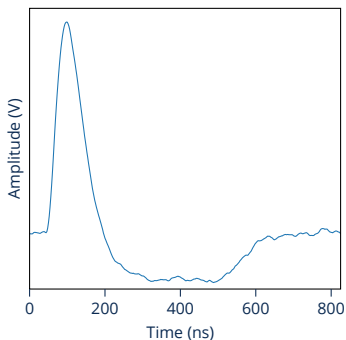
1. picking up signal and pile-up hits



The Readout Chain in AREUS



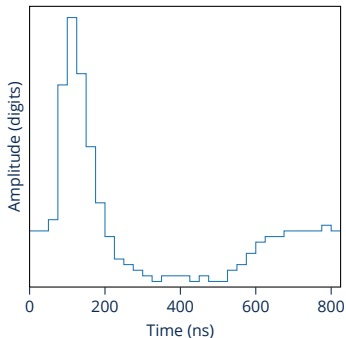
1. picking up signal and pile-up hits
2. adding analog pulses and electronics noise



The Readout Chain in AREUS



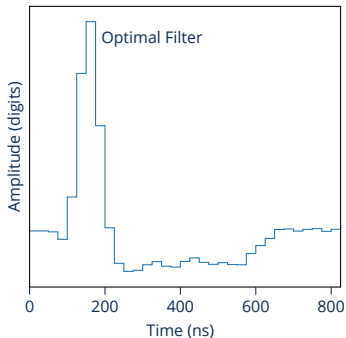
1. picking up signal and pile-up hits
2. adding analog pulses and electronics noise
3. quantization and quantization noise



The Readout Chain in AREUS



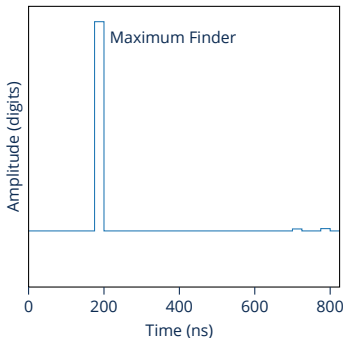
1. picking up signal and pile-up hits
2. adding analog pulses and electronics noise
3. quantization and quantization noise
4. signal processing with chain of digital filters



The Readout Chain in AREUS



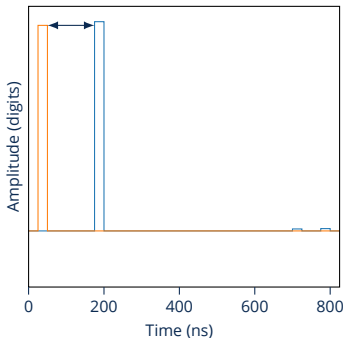
1. picking up signal and pile-up hits
2. adding analog pulses and electronics noise
3. quantization and quantization noise
4. signal processing with chain of digital filters



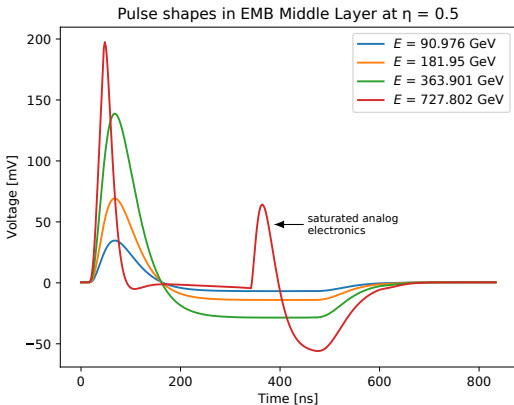
The Readout Chain in AREUS



1. picking up signal and pile-up hits
2. adding analog pulses and electronics noise
3. quantization and quantization noise
4. signal processing with chain of digital filters
5. comparison with input energy



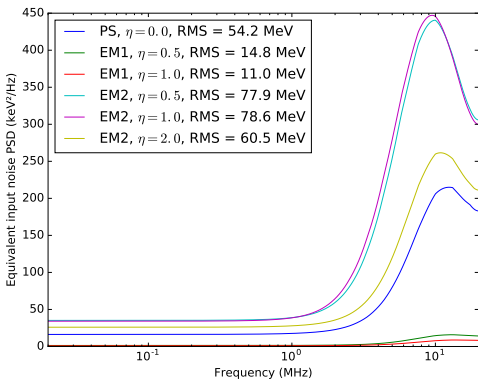
Pulse Shapes



- from SPICE electronics simulation
- considers saturation effects at high energies
- interpolation between pulses for different energies

Thermal Noise of Analog Electronics

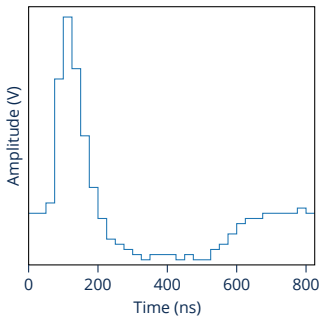
- three noise generators available:
 - white (~ 1)
 - pink ($\sim 1/f$)
 - arbitrary spectrum
- arbitrary spectra: automatic down-sampling to $f_N = \frac{1}{2}f_S = 20$ MHz



Examples for user-specified spectral densities passed to AREUS for noise simulation.

Digital Filters

- main task of AREUS simulation
- process continuous stream of digitized data
- for each bunch crossing, reconstruct energy and signal time



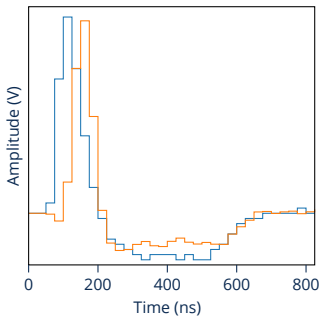
blue: unfiltered pulse

Digital Filters

- main task of AREUS simulation
- process continuous stream of digitized data
- for each bunch crossing, reconstruct energy and signal time

Optimal Filter

- based on χ^2 minimization of total noise
- ⇒ corrects pile-up on average



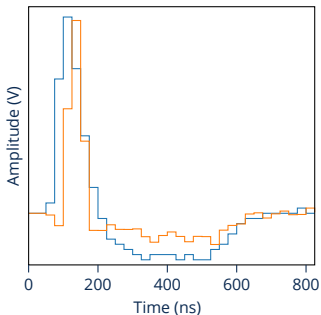
blue: unfiltered pulse
orange: OF response

Digital Filters

- main task of AREUS simulation
- process continuous stream of digitized data
- for each bunch crossing, reconstruct energy and signal time

Wiener Filter

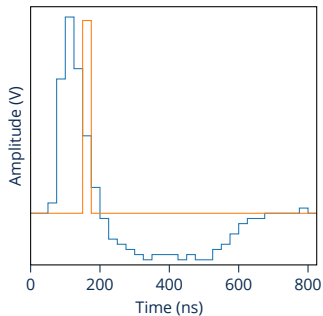
- deconvolves input with the expected pulse shape



blue: unfiltered pulse
orange: WF response

Digital Filters

- main task of AREUS simulation
- process continuous stream of digitized data
- for each bunch crossing, reconstruct energy and signal time

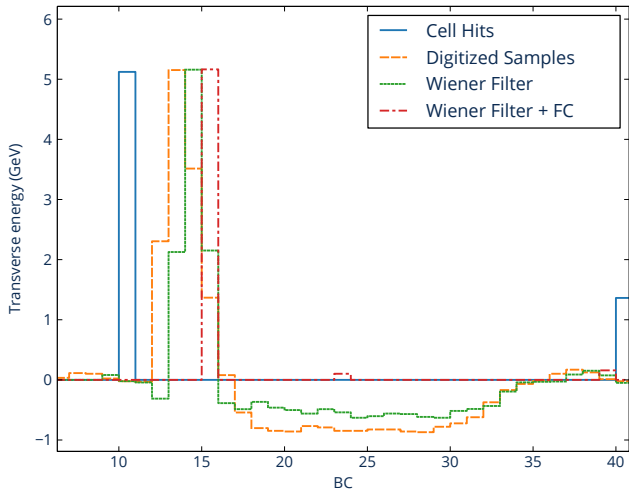


blue: unfiltered pulse
orange: WFFC response

WF + Forward Correction

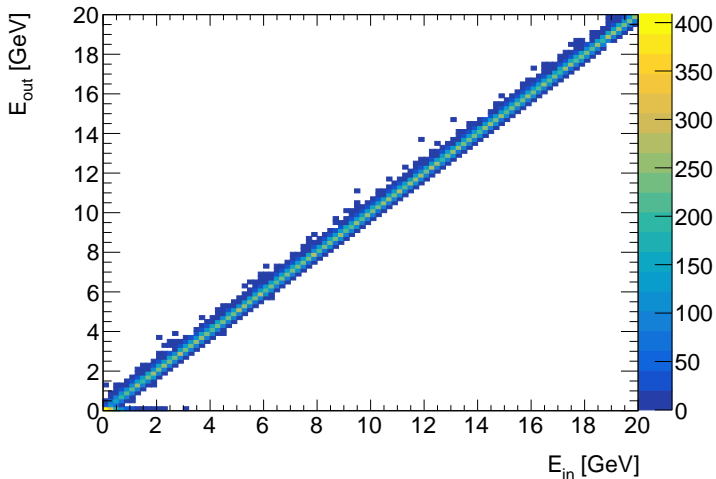
- subtracts detected pulses from subsequent samples
- ⇒ active pile-up correction event by event

AREUS Data Analysis



Samples sequences of simulated filters

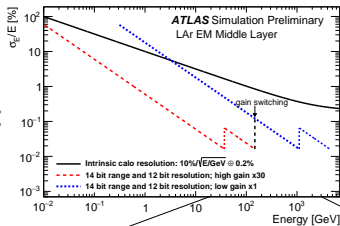
AREUS Data Analysis



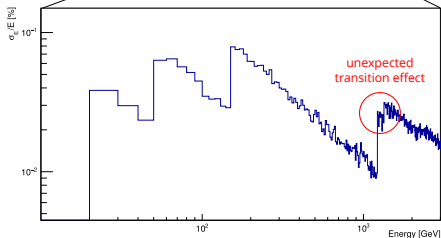
Comparison of deposited and reconstructed energy

AREUS Data Analysis

Calculation:

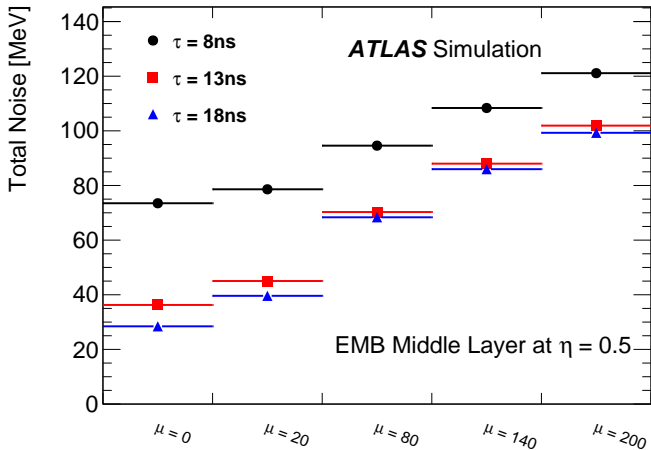


Detailed AREUS Simulation:



Effect of gain switching on the relative energy resolution

AREUS Data Analysis



Total noise as function of number of pile-up events μ
and of pulse-shaping time constant τ



AREUS

- \approx 51k lines of C++ code
- very modular design and configuration (XML file)
- design patterns: subject-observer, factory classes, ...
- external dependencies: **Boost**, **ROOT**
- bundled dependencies:
 - **ExprTK** for math parsing
 - **Google Test** for unit tests
 - **Splines** for interpolation
 - **VDT** for vectorized math


Conclusion

- valuable tool for LAr calorimeter electronics upgrade
- allows simulating long sequences of bunch crossings
- allows study of active signal filters
- flexible software design

Outlook

- more sophisticated filters
- interface to test-bench data processing
- further improvement of simulation accuracy (cross-talk between readout channels, ...)

References I

 *ATLAS liquid-argon calorimeter: Technical Design Report.*
Number CERN-LHCC-96-041. ATLAS-TDR-2 in
Technical Design Report ATLAS. Geneva, Jul 1996.
<https://cds.cern.ch/record/331061>.

 *ATLAS Liquid Argon Calorimeter Phase-I Upgrade Technical Design Report.*
Number CERN-LHCC-2013-017. ATLAS-TDR-022.
Geneva, Sep 2013.
<http://cds.cern.ch/record/1602230>.

 *ATLAS Phase-II Upgrade Scoping Document.*
Number CERN-LHCC-2015-020. LHCC-G-166.
Geneva, Sep 2015.
<https://cds.cern.ch/record/2055248>.

References II

-  *Technical Design Report for the Phase-II Upgrade of the ATLAS LAr Calorimeter.*
Number CERN-LHCC-2017-018. ATLAS-TDR-027.
Geneva, Sep 2017.
<https://cds.cern.ch/record/2285582>.
-  Maximilien Brice and Claudia Marcelloni.
View of the Detector ATLAS open.
<https://cds.cern.ch/record/1206860>, Feb 2007.
-  Joao Pequenaõ.
Computer generated image of the ATLAS Liquid Argon.
<https://cds.cern.ch/record/1095928>, Mar 2008.

References III



Joao Pequena.

Computer generated image of the whole ATLAS detector.

<https://cds.cern.ch/record/1095924>, Mar 2008.

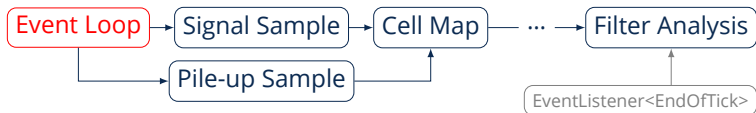
Backup: Technical Details

Config File

```
<?xml version="1.0"?>
<subject_config>
  <placeholders>
    <placeholder name="NOISEDB_PATH">$(DATADIR)/noise.root</placeholder>
    <placeholder name="PULSEDB_PATH">$(DATADIR)/pulses.root</placeholder>
    <placeholder name="ADC_BITWIDTH" default="12">$(ADC_BITWIDTH)</placeholder>
  </placeholders>
  <position_map type="LAR">map.xml</position_map>
  <subjects>
    <subject>
      <type>event_loop</type>
      <name>event_loop</name>
      <parameters><!--...--></parameters>
    </subject>
    <subject>
      <type>hit_sample</type>
      <name>sample_zee</name>
      <observe>event_loop</observe>
      <parameters><!--...--></parameters>
    </subject>
  <!--...-->
</subject_config>
```

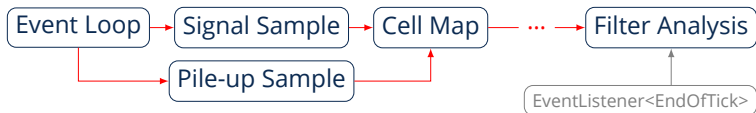
- contains all information for an AREUS run
 - input hit samples
 - pulse-shape & thermal-noise databases
 - chain of digital filters
 - parameters for filter analysis
- can define placeholders and read environment variables

Subjects and Observers



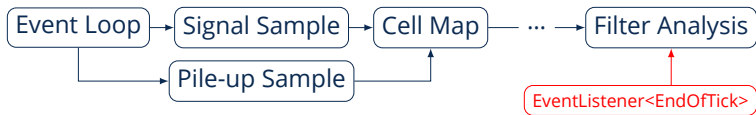
- simulation starts with a root subject

Subjects and Observers



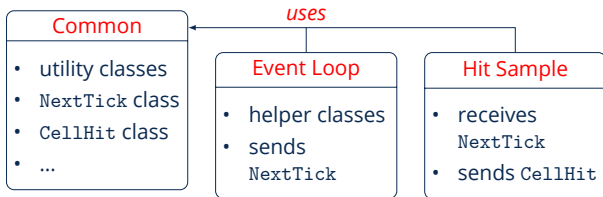
- simulation starts with a root subject
- recursive chain of notifications from subject to observer

Subjects and Observers



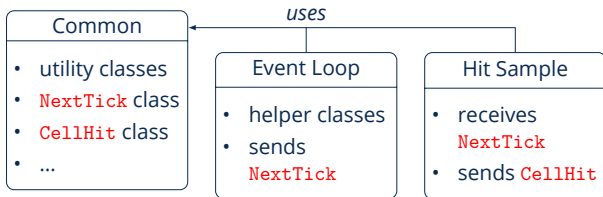
- simulation starts with a root subject
- recursive chain of notifications from subject to observer
- pre-/post-processing via inheritance from special class

Separation into Modules



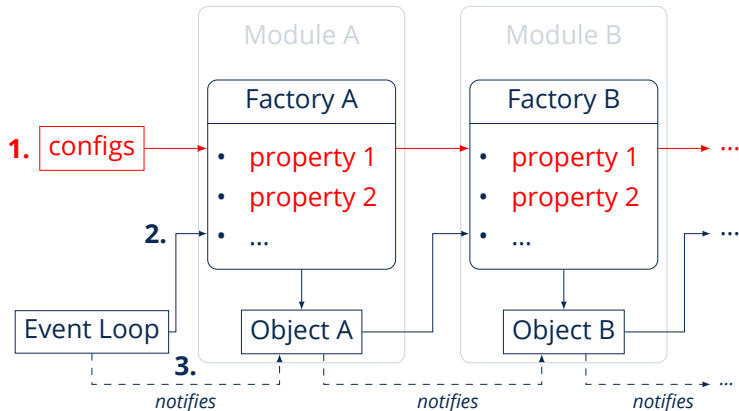
- all modules only see *Common* and themselves

Separation into Modules



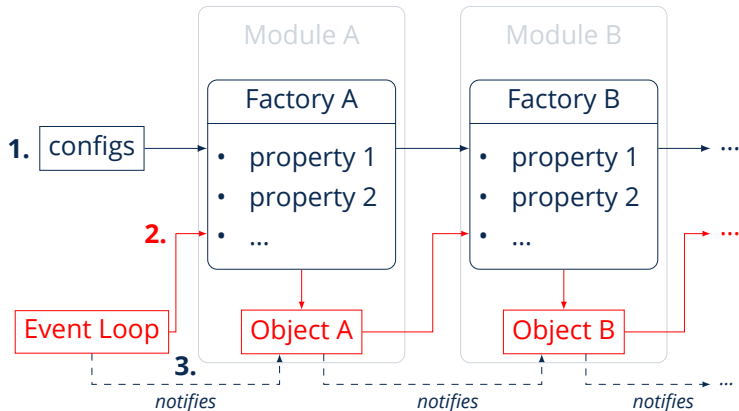
- all modules only see *Common* and themselves
- communication via *message* types

Factory Classes



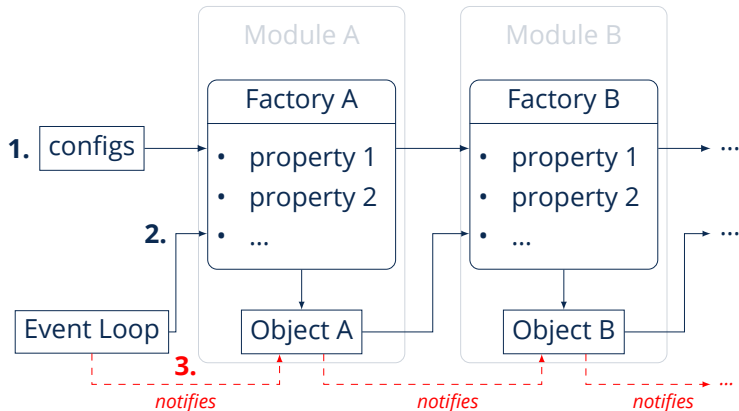
- extract and cache parameters from config file

Factory Classes



- extract and cache parameters from config file
- insert new objects into the subject–observer chain

Factory Classes



- extract and cache parameters from config file
- insert new objects into the subject-observer chain
- only used during initialization phase