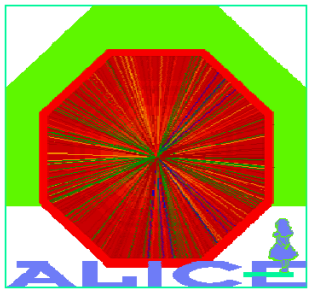


ALICE Online Data Quality Monitoring
MOOD Status

Filimon Roukoutakis

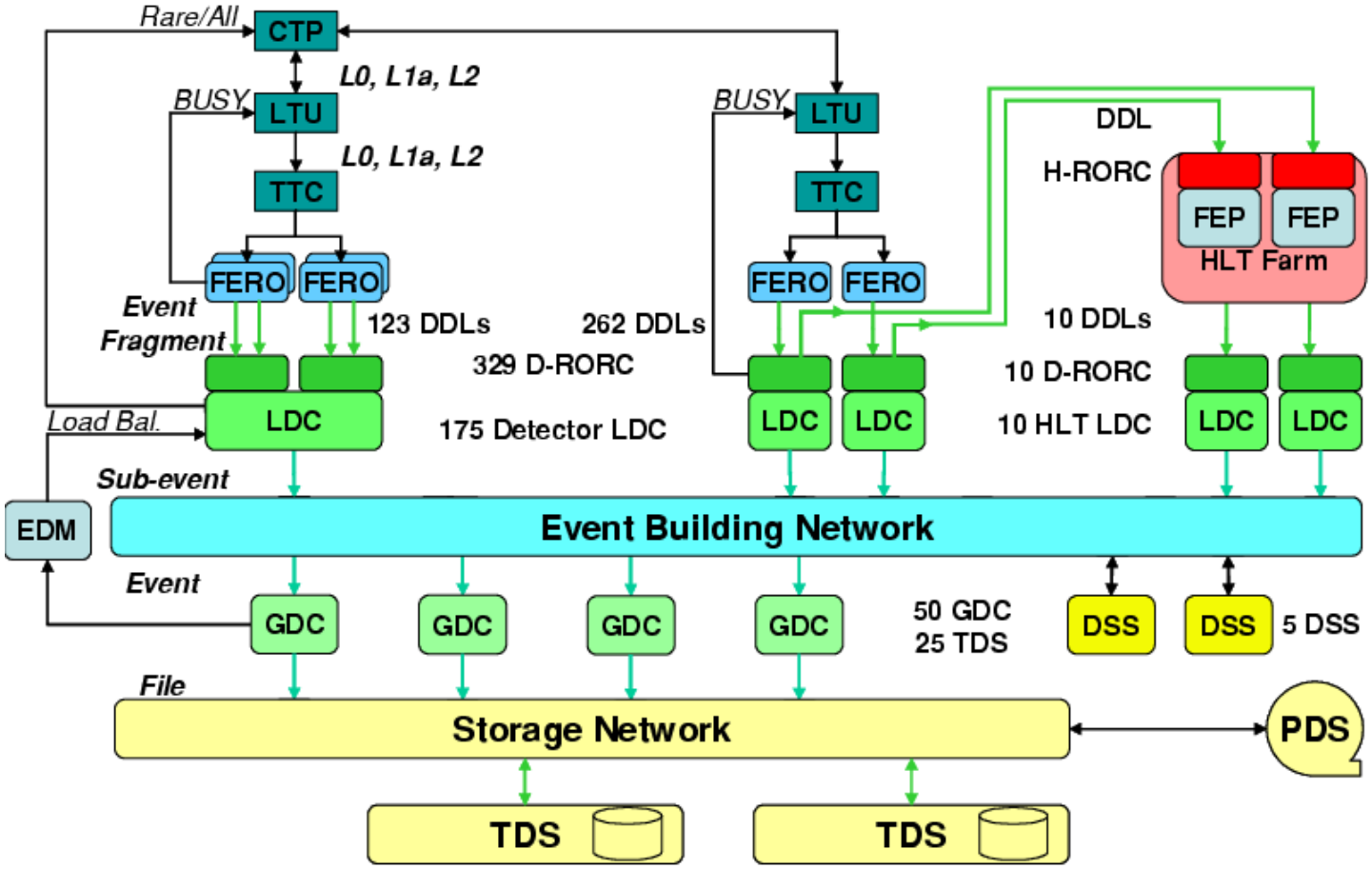
CERN PH-AID
&
University of Athens

ALICE Offline Week
CERN, 2/10/2006

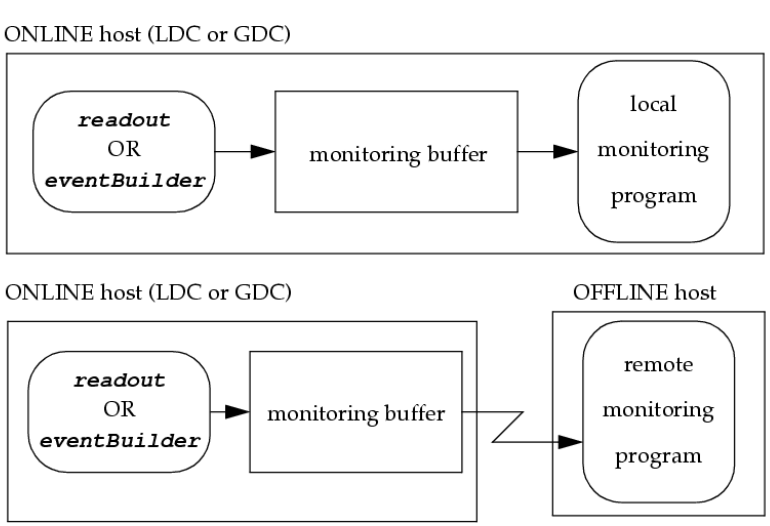
Outline

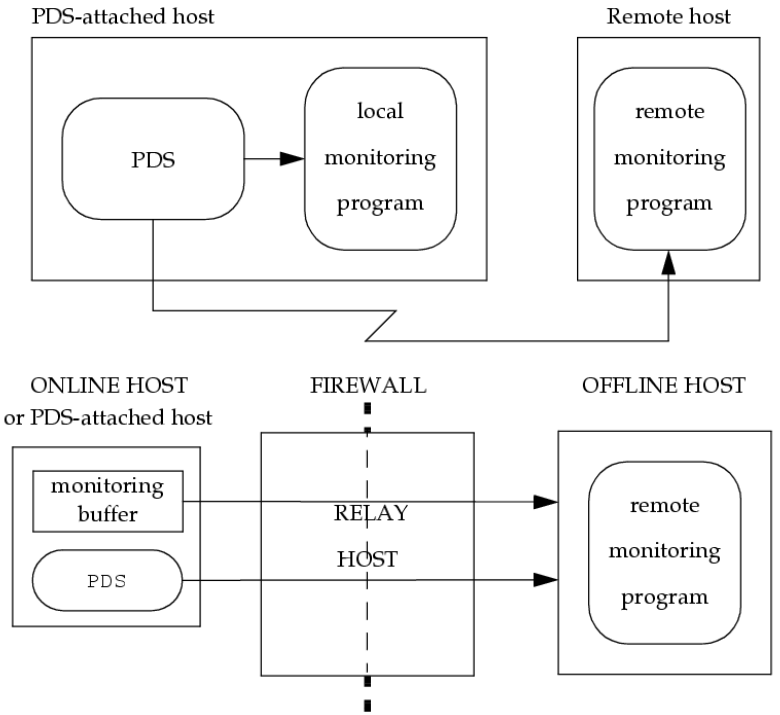
- ALICE DAQ Monitoring architecture
- MOOD
  - Introduction
  - Development Status/short term plans
    - Raw data decoding
    - Interface to Offline
    - Luminosity
- Automatic Data Quality Monitoring Plans
- Summary

ALICE DAQ Architecture



DATE Monitoring Scheme





- 3 policies: “ALL”, “YES”, “NO” possible for each type of event (“Physics”, “Calibration”, “SOR”, etc…)
- Local vs. Remote: More CPU vs. More Network
- Online vs. Offline:

Immediate feedback + Interference with DAQ

vs.

Delayed feedback + Extra storage layout consideration (Monitoring on TDS systems)

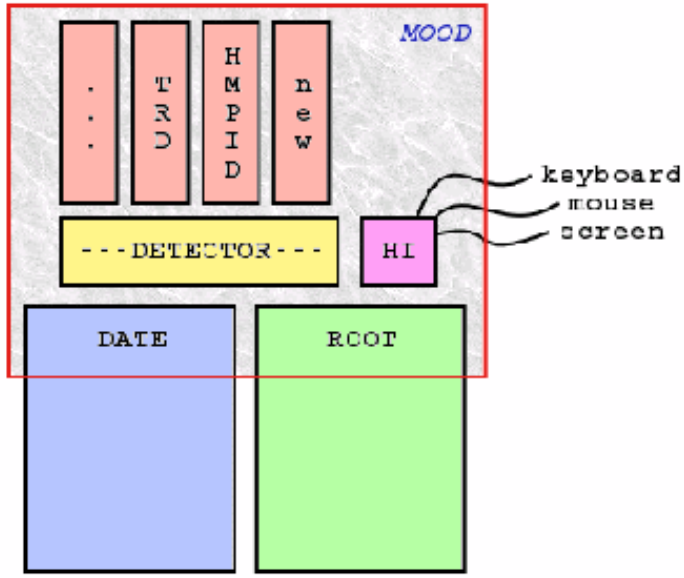
MOOD

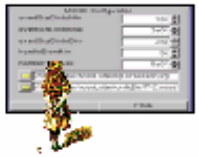
- Originally developed by Ozgur Cobanoglu for the DAQ group
- 2 Tasks: Online/Offline Data quality monitoring & Detector Debugging
- Written in C++, based on ROOT & DATE
- Runs under SLC 3/4, with DATE v5 and ROOT v5.12
- General framework has evolved, most of the core functionality (DATE interface, GUI) has been completely rewritten. The new scheme was introduced in release 5.00.01, 4 releases followed, now on 5.00.05.

MOOD Structure (1)

Structure

- MOOD runs as a monitoring client within DATE framework
- HI and event handling is based on ROOT libraries
- Data access is based on DATE monitoring libraries
- The tool has a modular structure; enhancement is easy



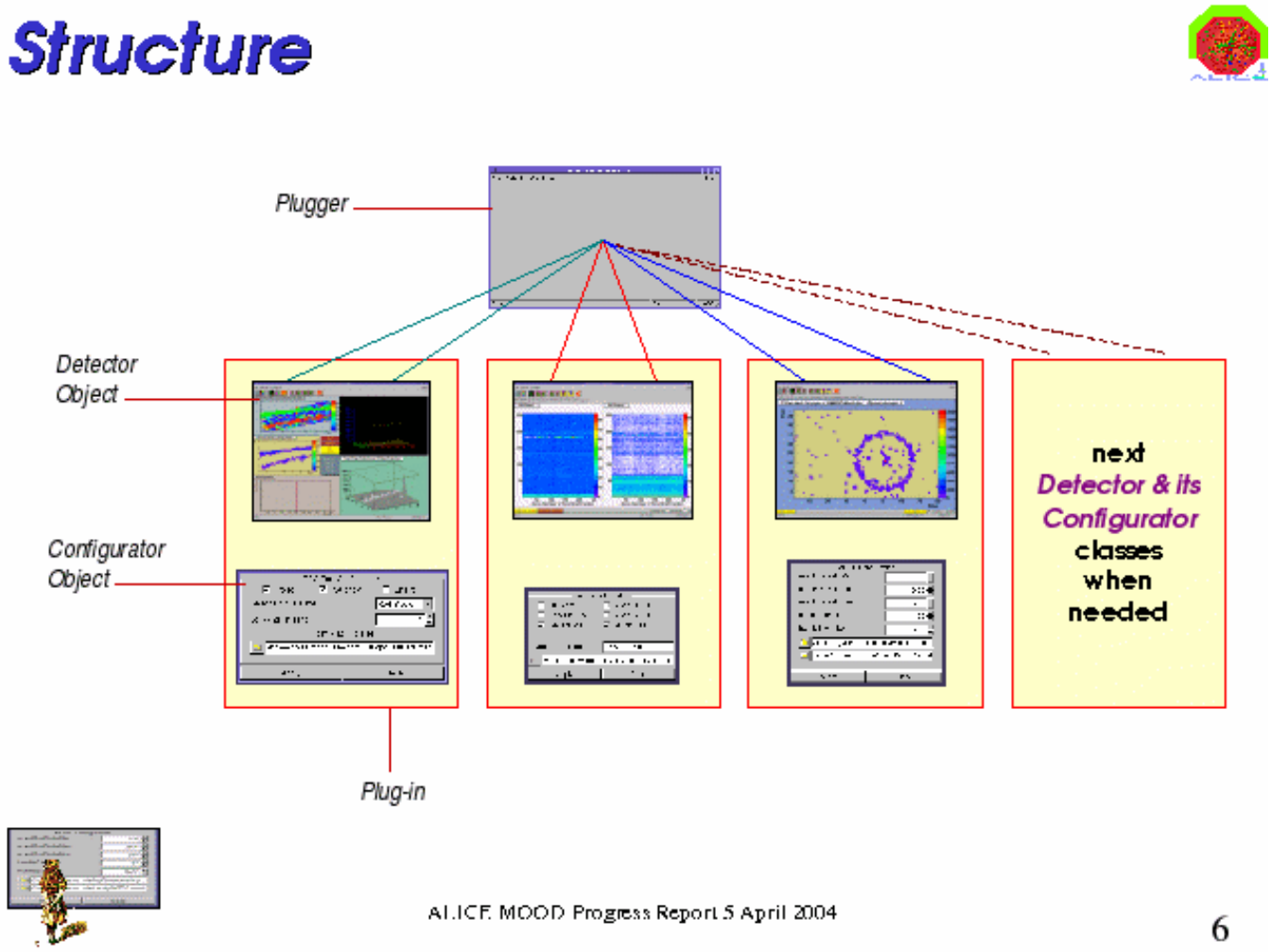


ALICE MOOD Progress Report 5 April 2004

5

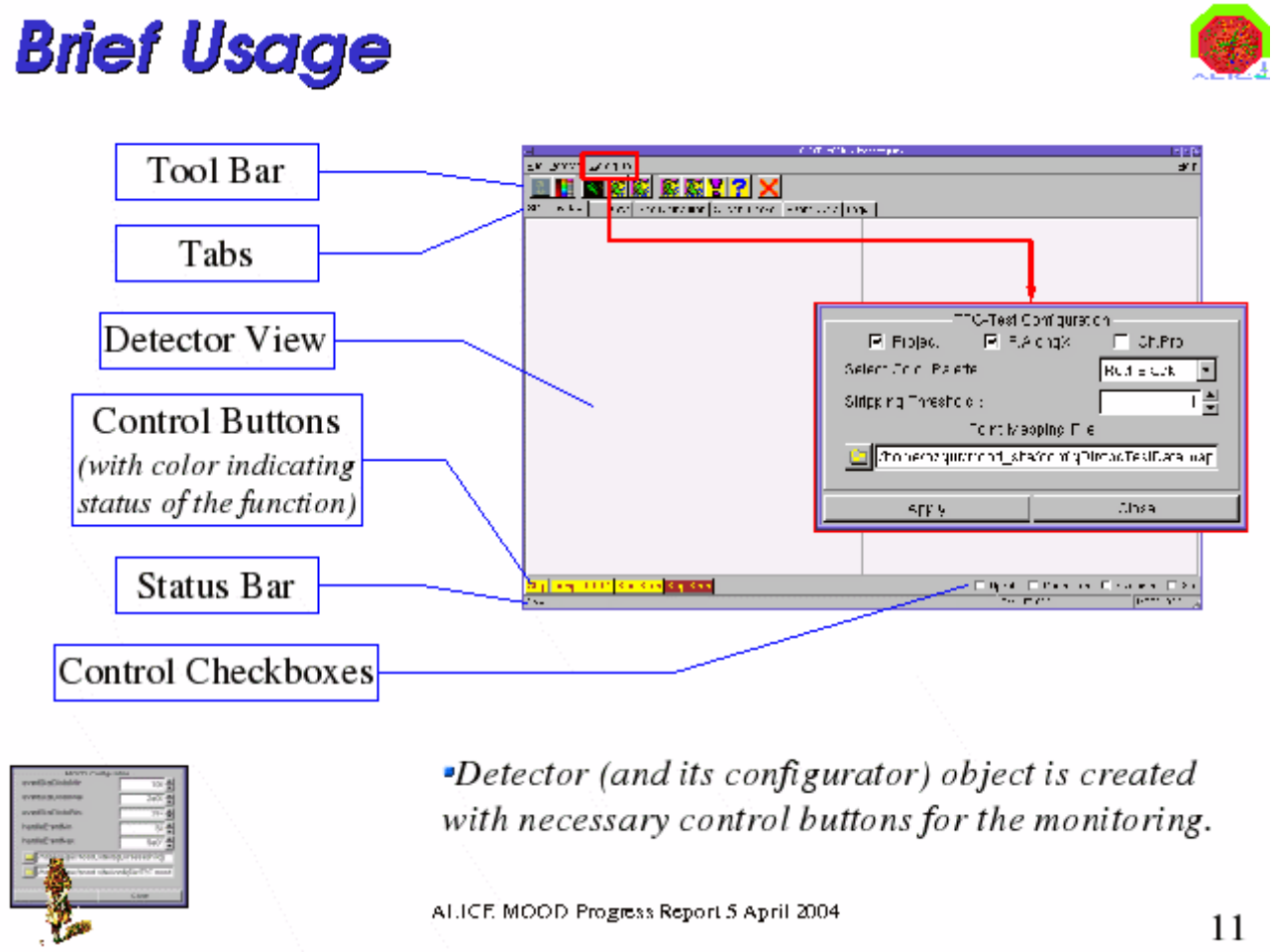
Picture taken from a presentation by Ozgur Cobanoglu

MOOD Structure (2)



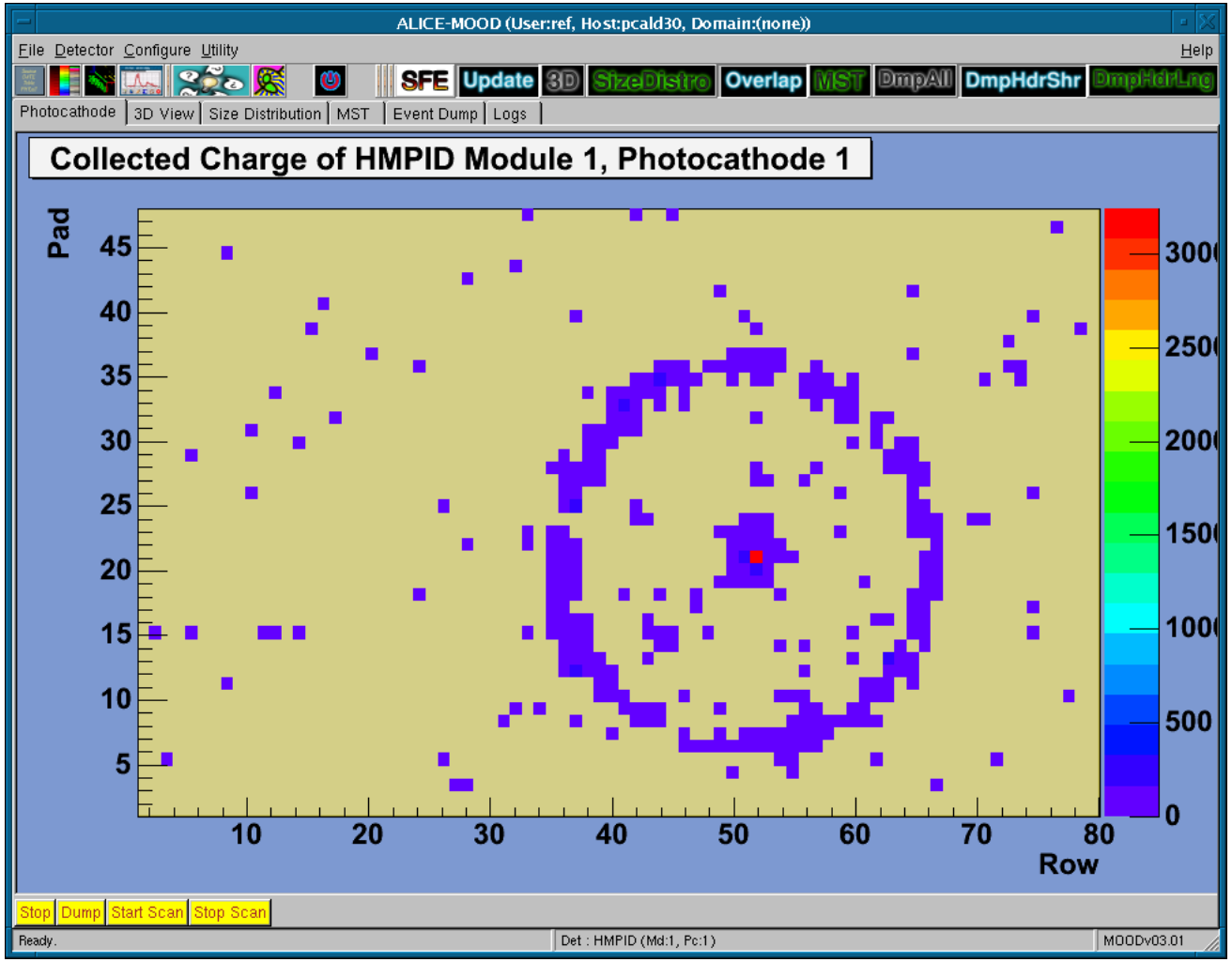
Picture taken from a presentation by Ozgur Cobanoglu

MOOD Main Screen

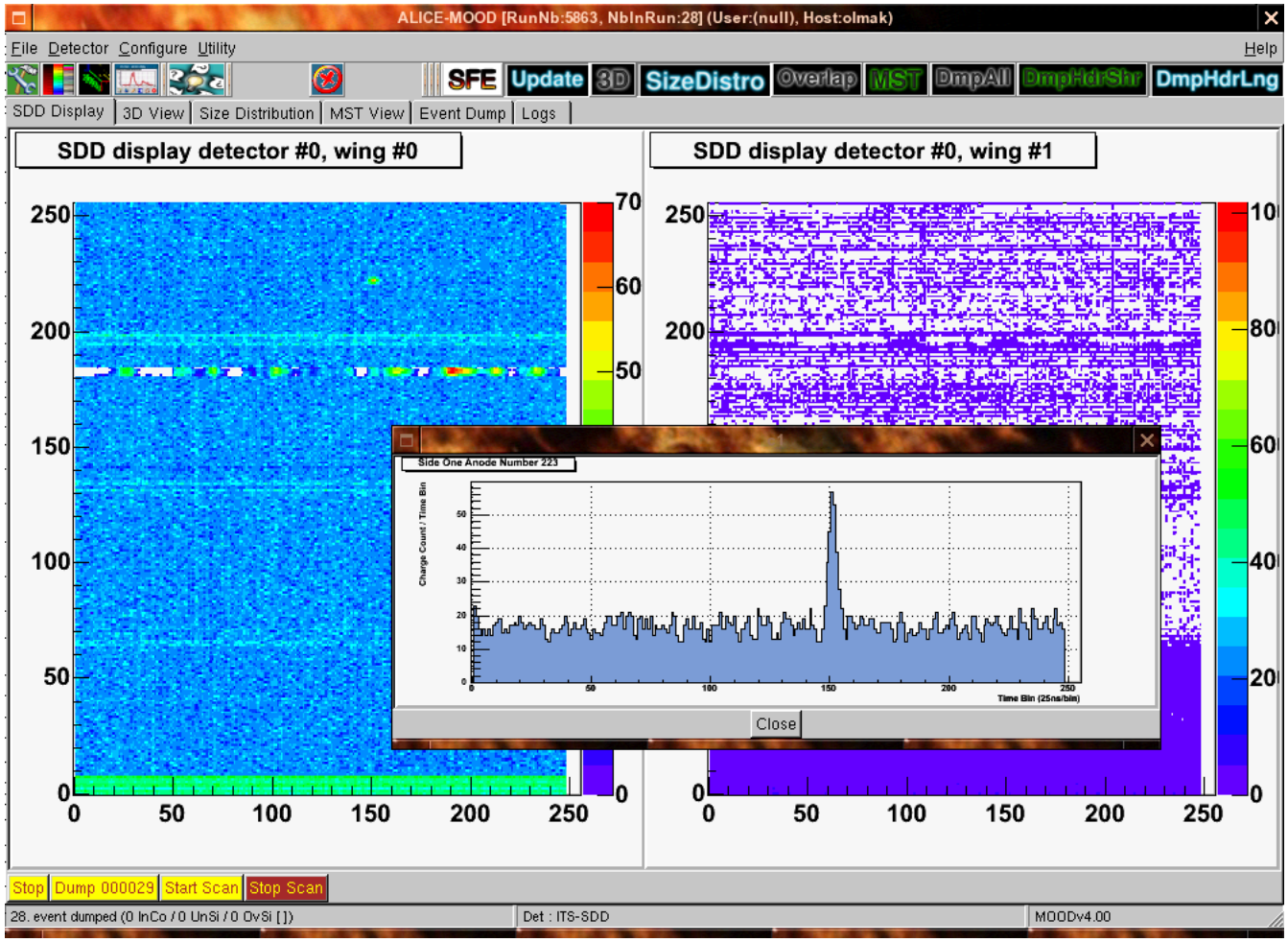


Picture taken from a presentation by Ozgur Cobanoglu

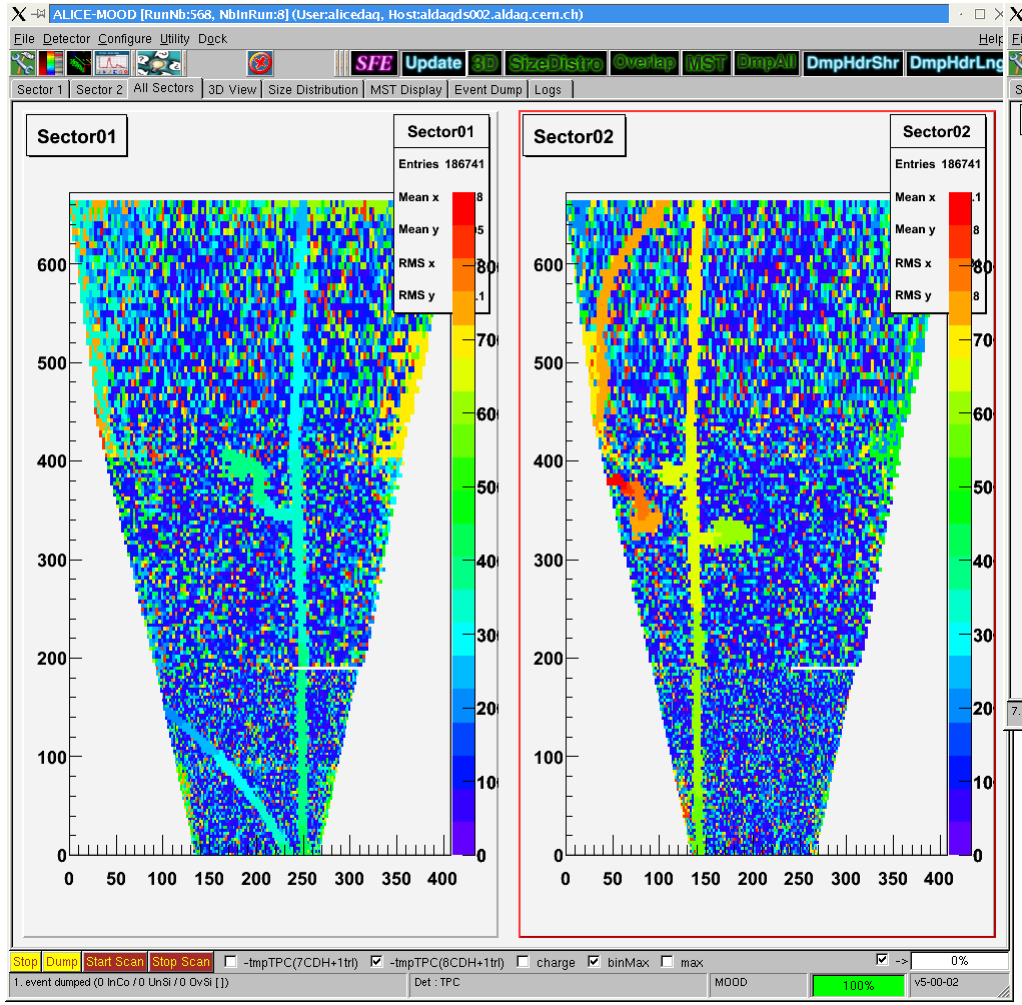
MOOD for HMPID

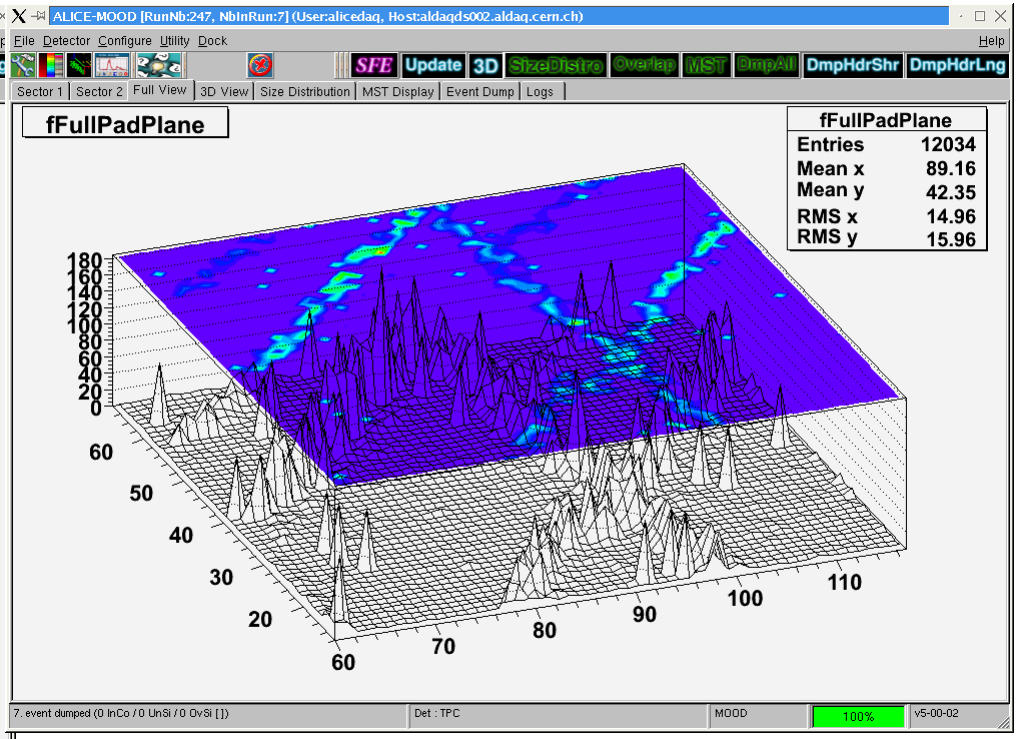


MOOD for SDD

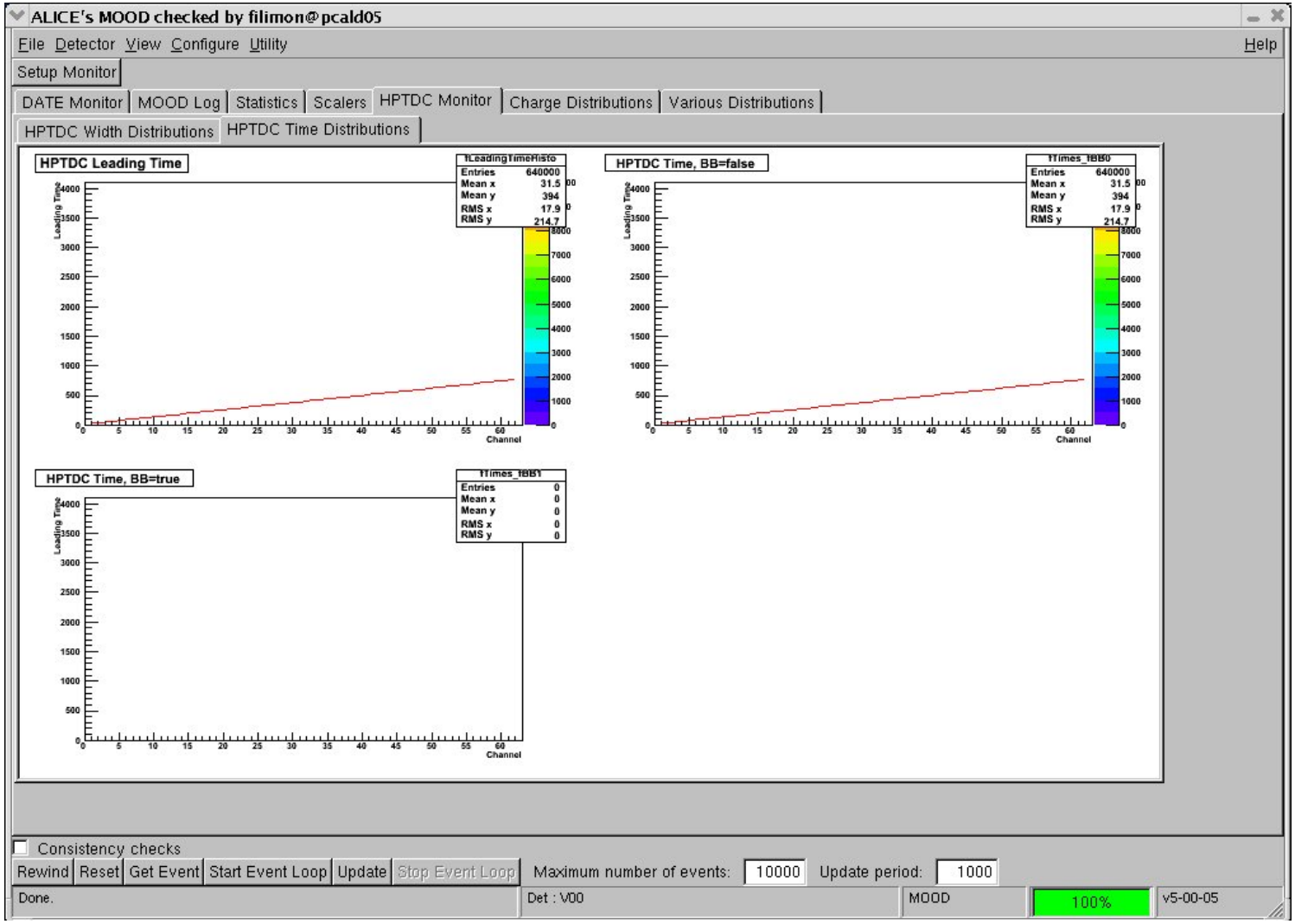

ALICE-MOOD [RunNb:5863, NbInRun:28] (User:(null), Host:olmak)
File Detector Configure Utility
Help
SFE
Update
3D
SizeDistro
Overlap
MST
DmpAll
DmpHdrShr
DmpHdrLng
SDD Display
3D View
Size Distribution
MST View
Event Dump
Logs
SDD display detector #0, wing #0
SDD display detector #0, wing #1
Side One Anode Number 223
Charge Count / Time Bin
Time Bin (25ns/bin)
Close
Stop
Dump 000029
Start Scan
Stop Scan
28. event dumped (0 InCo / 0 UnSi / 0 OvSi [])
Det : ITS-SDD
MOODv4.00

MOOD for TPC

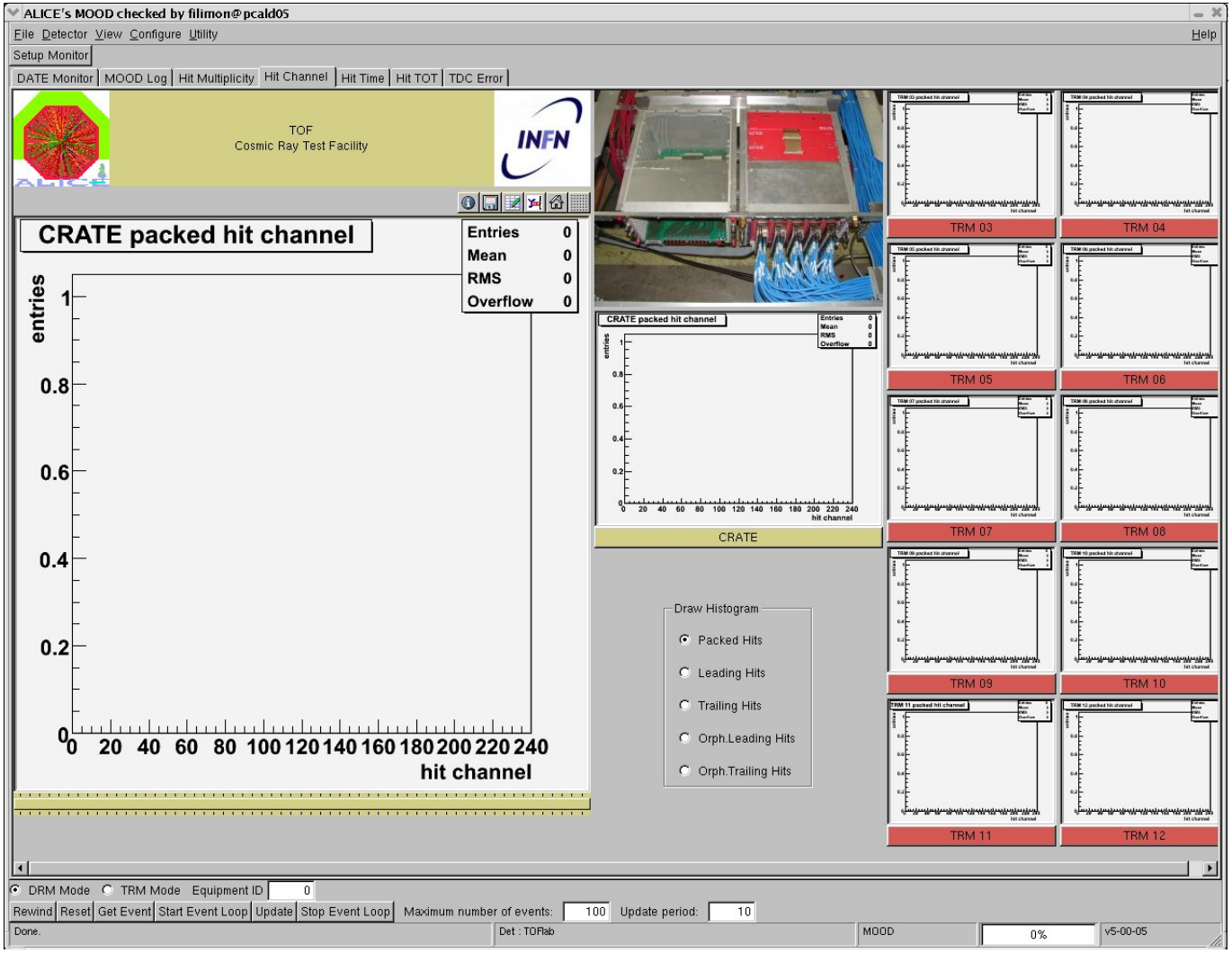

ALICE-MOOD [RunNb:568, NbInRun:8] (User:alicedaq, Host:aldaqds002.aldaq.cern.ch)
File Detector Configure Utility Dock
Sector 1 Sector 2 All Sectors 3D View Size Distribution MST Display Event Dump Logs
Sector01
Sector02
Entries 186741
Mean x
Mean y
RMS x
RMS y
Stop Dump Start Scan Stop Scan -tmpTPC(7CDH+1trl) -tmpTPC(8CDH+1trl) charge binMax max
1..event dumped (0 InCo / 0 UnSi / 0 OvSi [])
Det : TPC
MOOD
100%
v5-00-02


ALICE-MOOD [RunNb:247, NbInRun:7] (User:alicedaq, Host:aldaqds002.aldaq.cern.ch)
File Detector Configure Utility Dock
Sector 1 Sector 2 Full View 3D View Size Distribution MST Display Event Dump Logs
fFullPadPlane
Entries 12034
Mean x 89.16
Mean y 42.35
RMS x 14.96
RMS y 15.96
7. event dumped (0 InCo / 0 UnSi / 0 OvSi [])
Det : TPC
MOOD
100%
v5-00-02

MOOD for V0


ALICE's MOOD checked by filimon@pcald05
File Detector View Configure Utility
Help
Setup Monitor
DATE Monitor
MOOD Log
Statistics
Scalers
HPTDC Monitor
Charge Distributions
Various Distributions
HPTDC Width Distributions
HPTDC Time Distributions
HPTDC Leading Time
HPTDC Time, BB=false
HPTDC Time, BB=true
Entries 640000
Mean x 31.5
Mean y 394
RMS x 17.9
RMS y 214.7
Channel
Consistency checks
Rewind
Reset
Get Event
Start Event Loop
Update
Stop Event Loop
Maximum number of events: 10000
Update period: 1000
Done.
Det : V00
MOOD
100%
v5-00-05

MOOD for TOF


ALICE's MOOD checked by filimon@pcald05
File Detector View Configure Utility
Help
Setup Monitor
DATE Monitor
MOOD Log
Hit Multiplicity
Hit Channel
Hit Time
Hit TOT
TDC Error
TOF
Cosmic Ray Test Facility
INFN
CRATE packed hit channel
Entries 0
Mean 0
RMS 0
Overflow 0
entries
hit channel
CRATE
Draw Histogram
Packed Hits
Leading Hits
Trailing Hits
Orph.Leading Hits
Orph.Trailing Hits
TRM 03
TRM 04
TRM 05
TRM 06
TRM 07
TRM 08
TRM 09
TRM 10
TRM 11
TRM 12
DRM Mode
TRM Mode
Equipment ID
Rewind
Reset
Get Event
Start Event Loop
Update
Stop Event Loop
Maximum number of events:
Update period:
Done.
Det : TOFlab
MOOD
v5-00-05

MOOD Detector Status

1. SPD: Active development, new scheme (Henrik Tydesjo)
2. SDD: Previous scheme
3. SSD: Previous scheme (Slawomir Biegluk)
4. HMPID: Previous scheme (Antonello di Mauro)
5. TPC: Active development, new scheme (filimon), previous scheme (Ozgur Cobanoglu, Roland Bramm, Luciano Musa)
6. EMCAL: Not implemented, TPC RCU (David Sylvermyr from 12/2006)
7. PHOS: Not implemented, TPC RCU (filimon?)
8. TRD: Previous scheme (David Emschermann)
9. MUON Trigger: Active development, previous scheme (Valerie Barette)
10. MUON Tracker: Active development, new scheme (Guillaume Batigne)
11. TOF: Active development, new scheme (Roberto Preghenella), tested in TOF cosmic ray test facility
12. FMD: New scheme, under development (filimon)
13. V0: New scheme, implemented+further devel (filimon)
14. T0: New scheme, implemented+further devel (filimon)
15. ACCORDE: Not implemented (filimon?)
16. ZDC: Not implemented (filimon?)
17. PMD: Not implemented (filimon?)
18. Trigger: CTP decoder+GUI, access to CTP scalers through DIM (filimon)
19. DAQ: Online relavant quantities (event rates/sizes, trigger rates etc) (filimon)

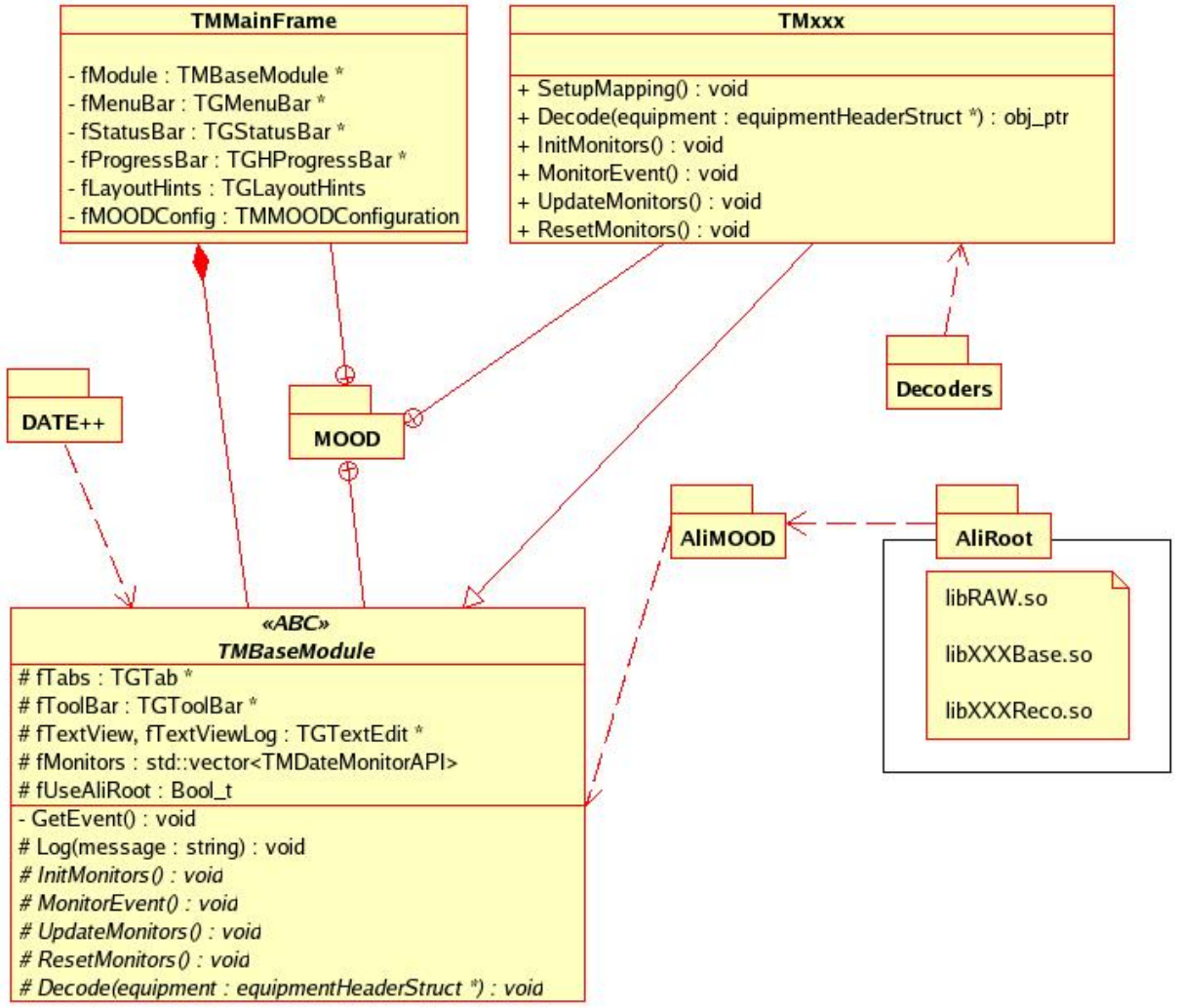
Useful MOOD Info

- Contact person: Filimon dot Roukoutakis at cern dot ch,
- Website: http://ph-dep-aid.web.cern.ch/ph-dep-aid/MOOD/
- http://hypernews.cern.ch group: “General MOOD Discussion”

MOOD framework features

- Monitoring elements (usually histograms) are laid on tabbed windows.
- Ability to monitor from several sources:
  - Online on LDC/GDC
  - Offline (files):
    - ALICE standard data format (ROOT) files on local disk, CASTOR or an http server (GDC events)
    - DATE raw format files on local disk or CASTOR (LDC/GDC)
- Multithreaded execution, GUI fully usable while monitoring
- Single Document Interface, MDI under consideration
- Signal-slot mechanism for handling GUI events
- Can be used as a user-friendly hex event dumper
- Ability to monitor a pre-specified number of events and update the screen automatically after a defined number of events or manually.
- XML configuration under designing.
- Link to AliRoot code (See next slides)

MOOD UML Class diagram



Online Decoding Library

- C/C++ STL based (no external dependencies, including ROOT)
- Equipment-level based: Decoder gets a pointer to an equipmentHeaderStruct of DATE and returns a pointer to struct/class with the decoded data.
- Decoder could be eventType based (different functions for decoding depending on the event type (Physics, Calibration, …)
- Mapping is detached from decoding (separate classes, could be friends)

Decoder Status

- DecoderRCU:
  - Build a 2-D “structure” with row number=Altro Hardware address, row content=Altro samples.
  - Currently implemented as plain dynamically allocated memory region, could go easily to std::map<hwAddress_t, std::valarray<uint16_t>>
  - Tested on TPC, FMD, PHOS raw data. Mapping ready for TPC, under development for FMD. Can be used by EmCal also.
  - To implement copy-less on-demand decoding of distinct Altro channels.
- DecoderDRM:
  - Container for DecoderTRM that is itself a container of pointers indexing the various 32-bit data words and headers/trailers.
  - Tested on TRM data from T0.
  - Can be used by TOF and T0.
  - To provide public std::const_iterators interface
- DecoderV00: Simple bitfields interpretation, no mapping needed.
- DecoderSPD: Under development+ already existing
- MUON Trigger and Tracker use AliRoot for decoding
- TRD, SSD, TOF, HMPID use specially developed decoders.

Luminosity Monitoring with MOOD

- Collect information by decoding mainly V0, T0 and SPD payload from the DAQ network.
- Combine with scaler information collected through a DIM client (under development) from CTP.
- Make calculations, visualize, store information on DCS database.

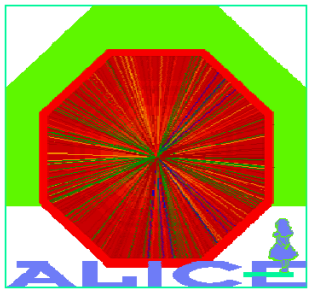
Interface to Offline (AliRoot)

- MOOD may be built with or without AliRoot present.
- Building with AliRoot is needed for monitoring offline data format files, through a “derootifier” addon library included.
- Interface to CDB for detector mapping information at SOR (Under consideration).
- Interface to On/Offline shuttle for transferring persistent monitor data (See dedicated slide-under discussion).
- Interface to Offline Decoding libraries? Discussion points: Implementation timeline, compatibility, organization of the efforts, online performance.
- Calibration/alignment information input?

Persistency of Monitoring Data

- The aim: Have an AliMSD.root for every run (Monitor Summary Data)
- The challenge: We have to take summary literally, no more than 10 GB of data per run in total for ALICE.
- The implementation: Two proposals
  1. Move the data at EOR from monitoring servers to TDS and then to CASTOR. Accessibility/tagging problem (MOOD needs to define metadata structures).
  2. Push the data at EOR to the shuttle to store on the Grid along with calib, align data.
- A weekly cache of data will be kept on monitoring servers.

What does the Offline group think?



Automatic Data Quality Monitoring
aka
ALICE’s future monitoring framework

Scope-General considerations

- Automatic DQM aims to collect monitoring data from various sources and take automated decisions on their quality with respect to various parameters and conditions and after comparison with reference values and plots.
- The proposed scheme is a client-server architecture:

  The server core(s) run on a number of machines that monitor the GDCs and LDCs directly and client machines in the ALICE Control Room (ACR) query and visualize the results. The reference values are stored in a central database.
- The collection scheme is planned to be the DATE monitoring library encapsulated under DATE++ class library.
- The processing backend is going to be written in C++ with minimal use of ROOT library only in data containers (e.g. histograms)
- The GUI will be written in C++ (qt3, ROOT) or Java

Scope-General considerations (II)

- The analysis of the data is going to be trigger- and run-conditions specific, e.g. check the multiplicity against expected values that differentiate themselves due to event centrality and checks of trigger rates in accordance to beam luminosity.
- Limited computing resources: Define and implement the minimum physics requirements (event size/channel occupancy, scaler readout, very simple cluster statistics).

Summary

- MOOD Plans:
  - Enhance the functionality and robustness of the existing application
  - Not yet implemented detector modules will follow the new unified scheme which enhances the modularity and reusability of code
  - Use MOOD as a test bed and debugger for the decoders before the introduction of Automatic DQM
- Automatic DQM Plans:
  - Collection of detector requirements has started (S. Mazzoni)
  - Finalize some design decisions based on the above requirements
  - Framework development timeline: Jan-Sep 2007, enhancements (automatic logic, ANN, full detector integration): Oct 2007-April 2008