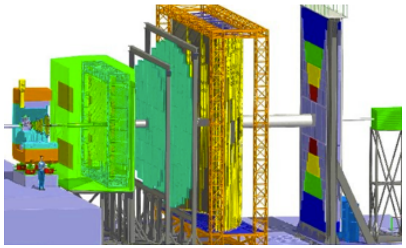
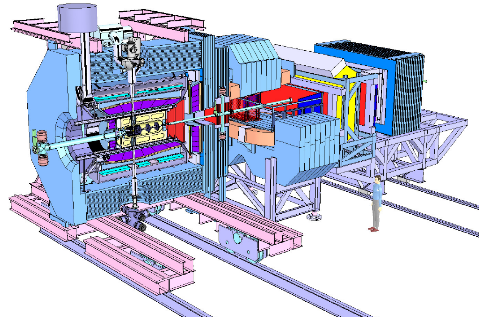
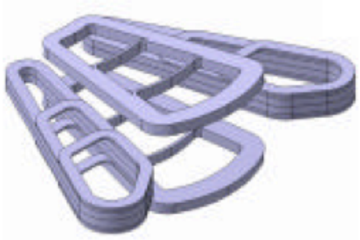
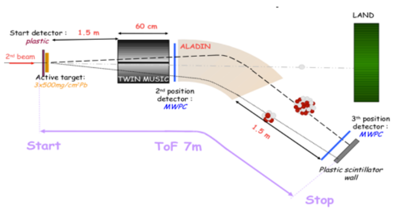
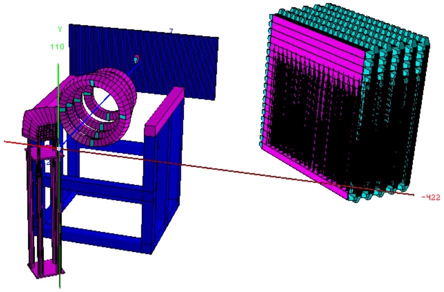
FairRoot Framework

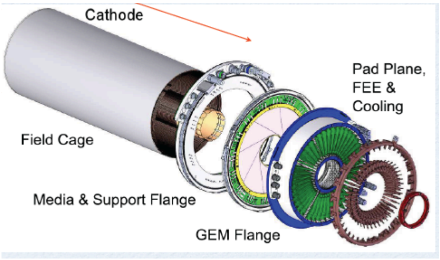


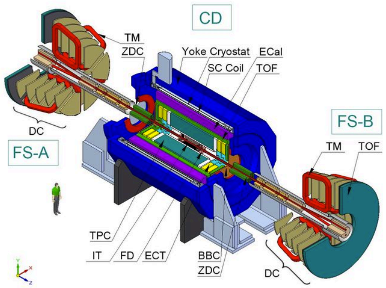


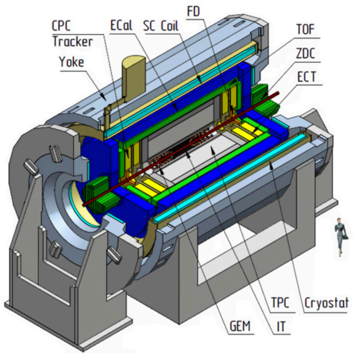


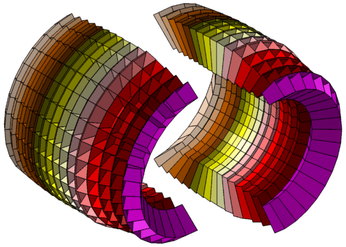












Mohammad Al-Turany (GSI-IT/CERN PH)

How it started?

CBM collaboration 2003

- We need simulations for the LOI
- We have no manpower for software
- Re-use existing software
- It has to be easy, fast, reliable, ..etc
- We need it yesterday

Geant3

AliRoot

FLUKA

PAW

VMC

Geant4

Pythia

ROOT

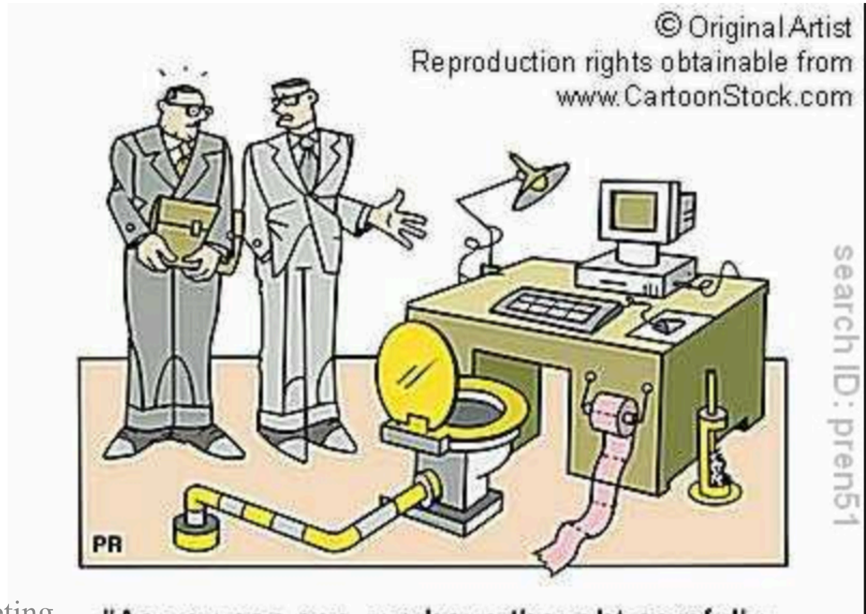
Urqmd

Go4

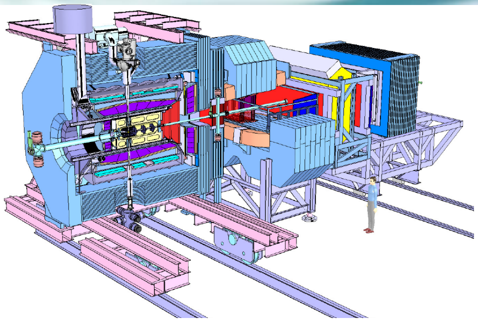
Hydra

How a framework can help?

- Allows physicists to concentrate on detector performance details, avoiding purely software engineering issues like storage, retrieval, code organization etc;
- Do not submerge into low-level details.
- Use pre-built and well-tested code for common tasks.



"As you can see, we have thought carefully about ways of improving staff



PANDA collaboration 2006

Motivations

• To investigate the CBMRoot framework, developed at GSI by D. Bertini and M. Al-Turany for the CBM Collaboration, for the full simulation and reconstruction of one of the Panda subdetectors as an alternative approach to the presently used framework.

• To develop and test the Stt reconstruction algorithms to have results on the Stt performances in short time. Optimization studies to define the STT parameters (# of layers, skew angle, thickness of layers…)

• In this presentation we will show the status of this work and give a report about our experience with CBMRoot.

Important points:
- Easy installation and portability
- ROOT environment (useful for non-BaBaR groups)
- Supported at GSI
- Virtual MonteCarlo (Geant3+Geant4+Fluka)

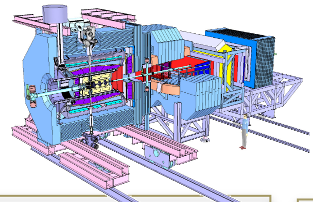
12-14 june 2006 A. Fontana, Panda Collaboration Meeting 21

FairRoot was officially created in 2006

- CBM and PANDA invest manpower in the core team
- The GSI decided to support the project
- Many motivated people from both experiments participate in the development of different features.

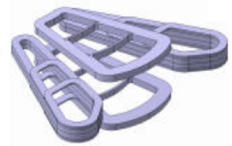
FairRoot

Start testing the VMC concept for CBM

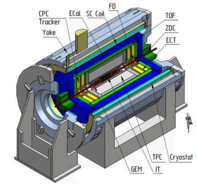


Panda decided to join->

FairRoot: same Base package for different experiments

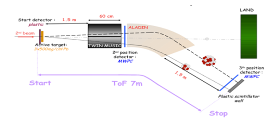


R3B joined

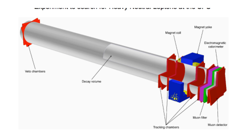


EIC (Electron Ion Collider BNL)

EICRoot



SOFIA (Studies On Fission with Aladin)



SHIP - Search for HIdden Particles

2004 | 2006 | 2010 | 2011 | 2012 | 2013 | 2014

First Release of CbmRoot

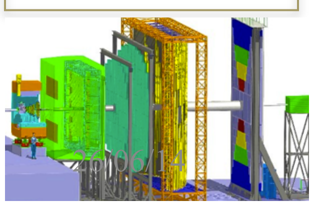
MPD (NICA) start also using FairRoot

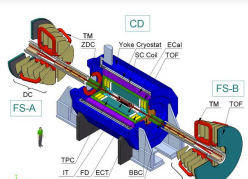
ASYEOS joined (ASYEOSRoot)

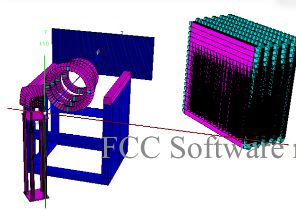
GEM-TPC seperated from PANDA branch (FOPIRoot)

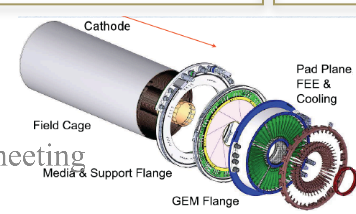
CALIFA (CALorimeter for the In Flight detection of γ rays and light charged pArticles)

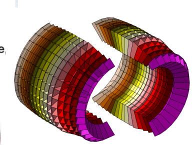
ENSAR-ROOT Collection of modules used by structural nuclear phsyics exp.



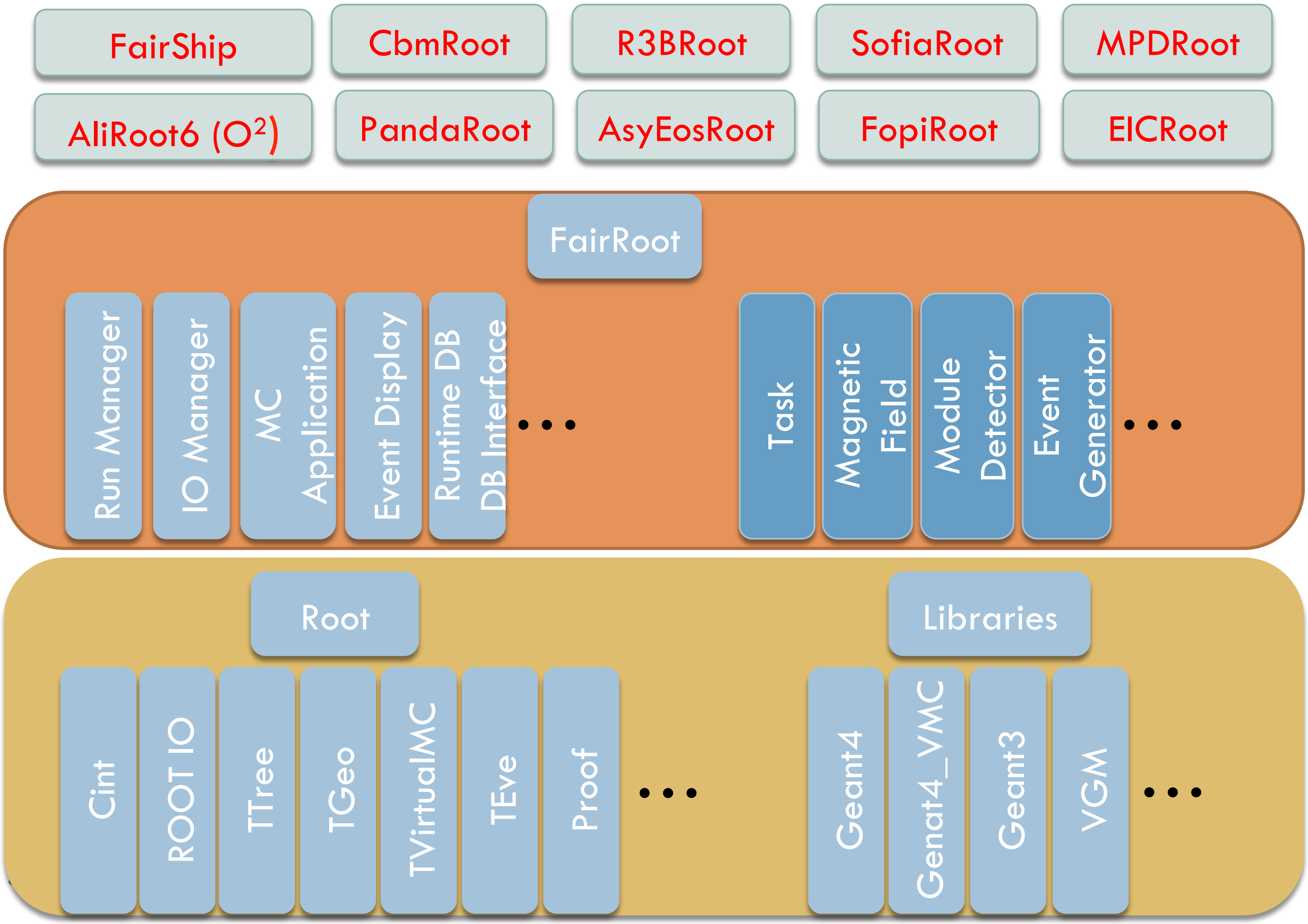








Code organization of FairRoot



FairRoot in nutshell

- An open source project (LGPL V3) available from GitHub
  https://github.com/FairRootGroup/FairRoot
- Simulation-, Reconstruction-, and Analysis-Framework (not only) for the FAIR experiments
- 2003 started as 2 person project for the CBM experiment
- 2014 ≈ 10 experiments use FairRoot as base for their developments
- Core team of 5 Developers (3.5 FTE)
- Many people contribute to make the project a success

Design

- Re use existing software and tools (use standards)
- Code should run on all platforms
- Framework should be
  - Easy to install
  - Easy to use
  - Should allow fast development cycles
  - Flexible to easily change experimental setup
  - Extensible for new developments

Easy to install

- Provide packages with all dependencies (ROOT, Geant3, Geant4, CMake, Boost, …) plus scripts for automatic installation on all systems

  https://github.com/FairRootGroup/FairSoft/

- Use CMake as build system and CTest/CDash for automatic testing and QA

  http://cdash.gsi.de/CDash/index.php?project=FairRoot

- Works on Mac OSX and many Linux derivatives (Debian, Ubuntu, Suse, Fedora, Scientific Linux), probably on many more which are not tested by us

Easy start for beginners

- Simulation and reconstruction examples are available

https://github.com/FairRootGroup/FairRoot/tree/dev/example

- Template for creating new detector setups are delivered with a rename script (Detector classes, data classes , ...etc can be created in seconds)

https://github.com/FairRootGroup/FairRoot/tree/dev/templates

(Also for reconstruction/analysis tasks or even whole projects)

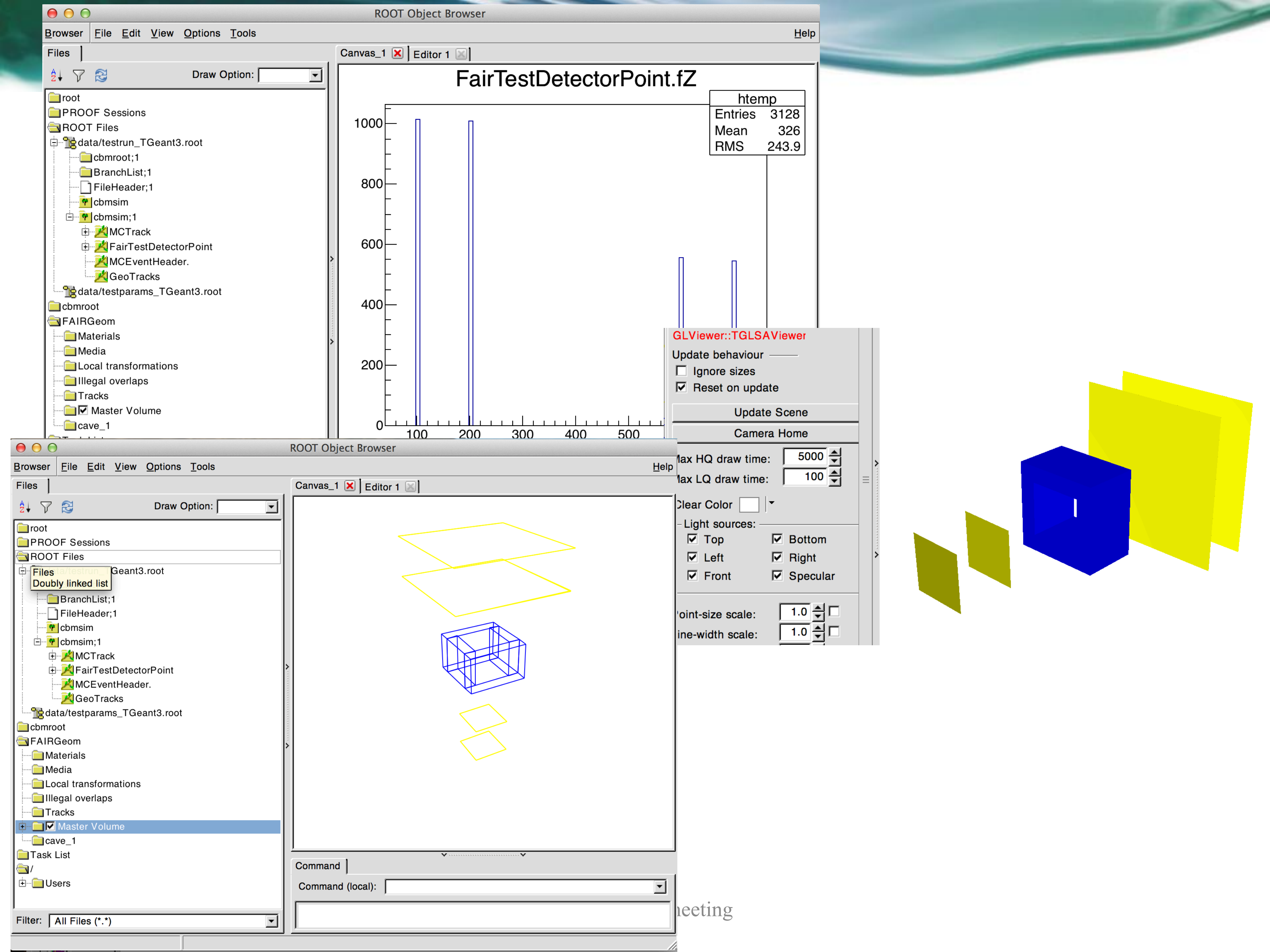
https://github.com/FairRootGroup/FairRoot/tree/dev#using-the-project-template

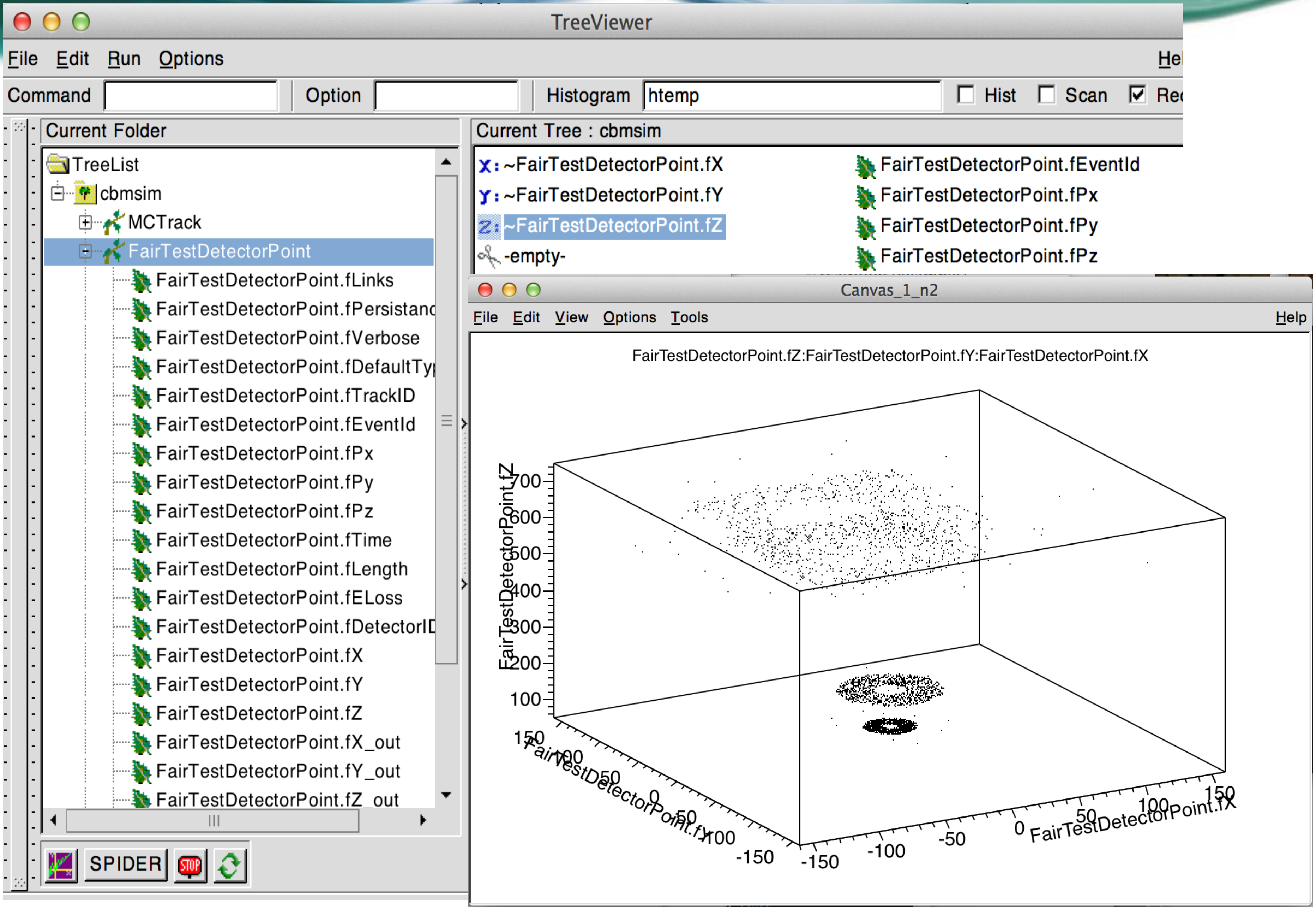
Easy start for beginners

- Geometry can be defined directly in code, in simple ASCII format, or taken from ROOT files (TGeo Format)

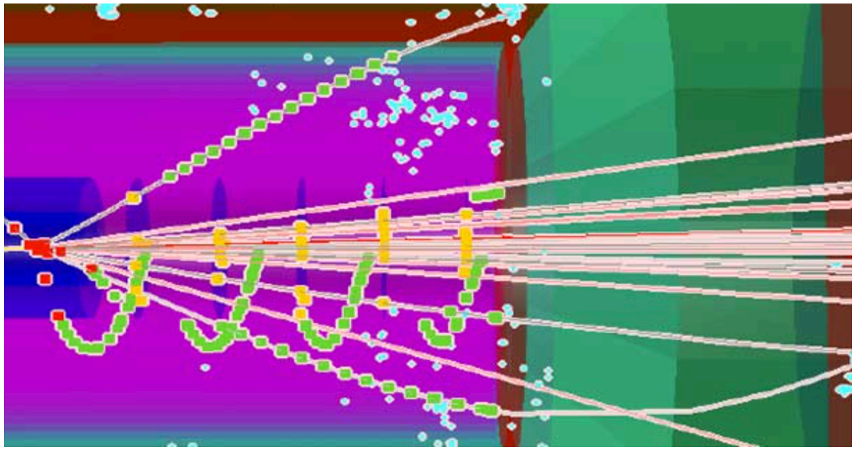
- Simulation output are simple root files that can be read by plain ROOT
  - Simple Ntuple analysis can be done in root macro (or even graphically: TTreeViewer)

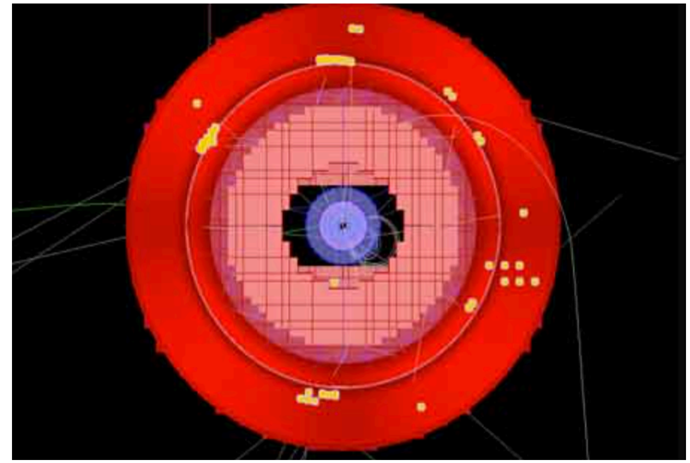
- Tools to visualize the geometry and the tracks are immediately available to the users

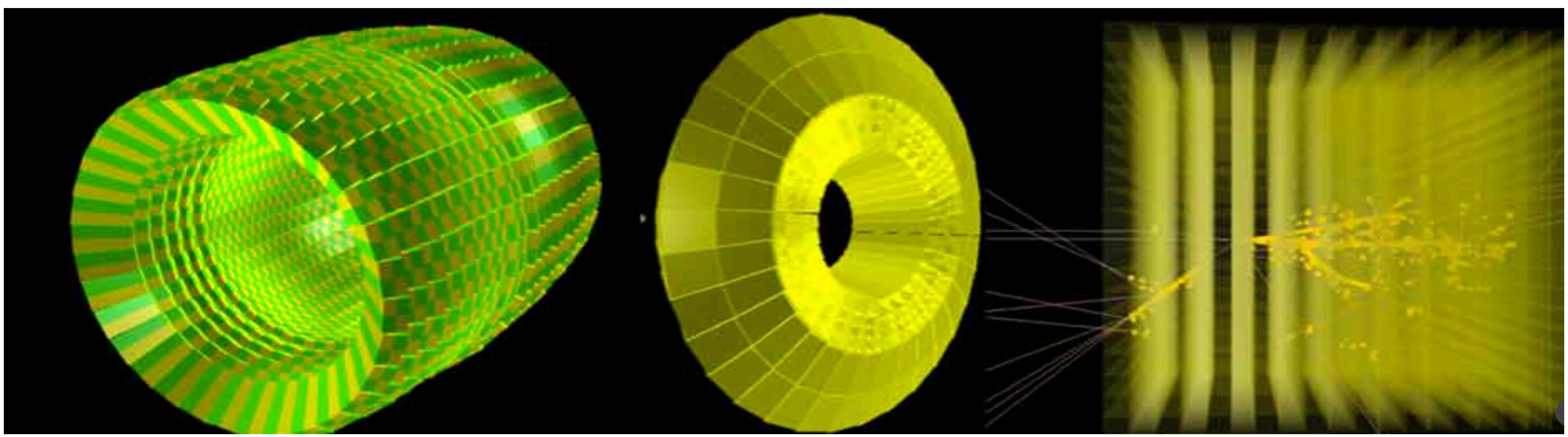

ROOT Object Browser
Browser File Edit View Options Tools Help
Files
Draw Option:
root
PROOF Sessions
ROOT Files
data/testrun_TGeant3.root
cbmroot;1
BranchList;1
FileHeader;1
cbmsim
cbmsim;1
MCTrack
FairTestDetectorPoint
MCEventHeader.
GeoTracks
data/testparams_TGeant3.root
cbmroot
FAIRGeom
Materials
Media
Local transformations
Illegal overlaps
Tracks
Master Volume
cave_1
Canvas_1
Editor 1
FairTestDetectorPoint.fZ
htemp
Entries 3128
Mean 326
RMS 243.9
1000
800
600
400
200
0
100
200
300
400
500
GLViewer::TGLSAViewer
Update behaviour
Ignore sizes
Reset on update
Update Scene
Camera Home
Max HQ draw time: 5000
Max LQ draw time: 100
Clear Color
Light sources:
Top
Bottom
Left
Right
Front
Specular
Point-size scale: 1.0
Line-width scale: 1.0
Files
Doubly linked list
Geant3.root
Task List
/
Users
Filter: All Files (*.*)
Command
Command (local):


TreeViewer
File Edit Run Options Hel
Command Option Histogram htemp Hist Scan Rec
Current Folder
TreeList
cbmsim
MCTrack
FairTestDetectorPoint
FairTestDetectorPoint.fLinks
FairTestDetectorPoint.fPersistanc
FairTestDetectorPoint.fVerbose
FairTestDetectorPoint.fDefaultTy
FairTestDetectorPoint.fTrackID
FairTestDetectorPoint.fEventId
FairTestDetectorPoint.fPx
FairTestDetectorPoint.fPy
FairTestDetectorPoint.fPz
FairTestDetectorPoint.fTime
FairTestDetectorPoint.fLength
FairTestDetectorPoint.fELoss
FairTestDetectorPoint.fDetectorID
FairTestDetectorPoint.fX
FairTestDetectorPoint.fY
FairTestDetectorPoint.fZ
FairTestDetectorPoint.fX_out
FairTestDetectorPoint.fY_out
FairTestDetectorPoint.fZ_out
SPIDER
Current Tree : cbmsim
X: ~FairTestDetectorPoint.fX
Y: ~FairTestDetectorPoint.fY
Z: ~FairTestDetectorPoint.fZ
-empty-
FairTestDetectorPoint.fEventId
FairTestDetectorPoint.fPx
FairTestDetectorPoint.fPy
FairTestDetectorPoint.fPz
Canvas_1_n2
File Edit View Options Tools Help
FairTestDetectorPoint.fZ:FairTestDetectorPoint.fY:FairTestDetectorPoint.fX
FairTestDetectorPoint.fZ
FairTestDetectorPoint.fY
FairTestDetectorPoint.fX

Event Display based on ROOT EVE package







Runs out of the box for track visualization in simulation

Flexibility

- Define run configuration at runtime
  - Use Root macros to define the experimental setup or the tasks for reconstruction/analysis
  - Use Root macros to set the configuration (Geant3, Geant4, …)
- No executable
  - Use plug-in mechanism from Root to load libraries only when needed
- No fixed simulation engine
  - Use different simulation engine (Geant3, Geant4, …) with the same user code (VMC)

Flexibility

- No fixed output structure
  - Store only the registered data classes to file
  - Use a dynamic event structure based on Root TFolder and TTree which is created automatically
  - Data output possible after each step
  - Different data levels can be connected via “Friend mechanism” in ROOT
- Simulation and reconstruction can be done in one go or in several steps
- Parameter handling
  - Use the parameter manager developed for the HADES experiment
  - Decouple parameter handling in FairRoot from parameter storage
  - Parameter manager IO to/from
    - ASCII files
    - Root files
    - Database

Simulation:

- The framework deliver a set of base classes, which has to be specialized by the user to describe his detector. i.e:
  - Detector
  - Module
  - MCPoints
  - Magnetic field
  - …etc
- The IO is handled completely by the framework
- Simulation is steered and configured via root macros

Fast Simulation

- Fast Simulation **reads same VMC stack** to get particles, therefore **all** event generators are supported with no changes to be done to the Fast Sim codes;
- Fast Simulation may **use acceptance parametization calculated from Full Simulation** or fast helix approximation for charged particles;

- Fast Simulation also works as a converter from ASCII to ROOT for event generators

Fast Simulation

- The same application, just different configuration:
  - – Event generators just push the event into the stack, no transport is taking place
  - –Detector response is presented as FairTasks (TTask)
  - –The output has the same form as full simulation

Testing and building system

- Cmake
  - Create Makefiles (and/or project files) for different platforms.
  - Test support.
  - Large user base assures support.
- CDash to handle data created with CMake
  - PHP framework
  - MySQL database
- Both tools are open source.

Time based simulation in FairRoot (Continues Read-Out)

- How do events overlap?
- In Detectors:
  - Sensor elements are still blocked from previous hits
  - Electronic is still busy
  - Hits too close in time cannot be distinguished
  - ...
- Special problem for CBM and PANDA:
  - Continuous beam with Poisson statistics (?) ➔ many events with short time between them
  - No hardware trigger
  - Complex event reconstruction
  - ➔ Necessary to simulate data stream as realistic as possible

Continues Read-Out : Implementation

- FairWriteoutBuffer is Special buffer to store detector data between different events
- You give the data you want to store an absolute time window this data is active in your detector and can influence later events.
- If the same detector element is hit a second time the data is modified.
- This is an abstract base class where you have to inherit from

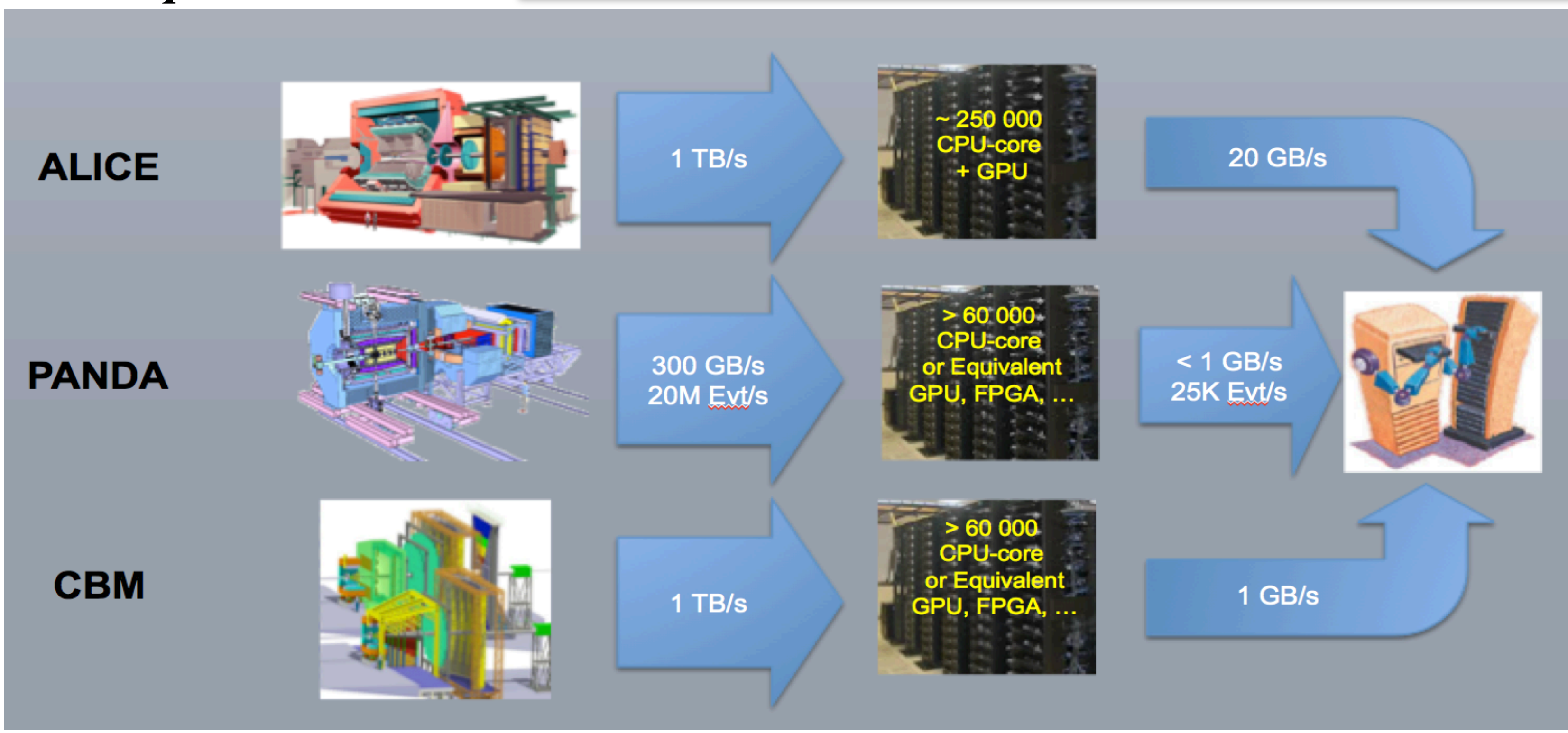
Continues Read-Out : Reading back data

- FairRootManager has new reading algorithms, which make it possible to use the event wise implemented tasks to run on such data streams
- Different algorithms available to extract data:
  - All data up to a given time
  - All data in a time window
  - All data between time gaps of a certain size
- Other algorithms can be (easily) implemented

The ALFA project

A common concurrency framework for ALICE and FAIR experiments

How to distribute the processes?
How to manage the data flow?
How to recover processes when they crash?
How to monitor the whole system?
……

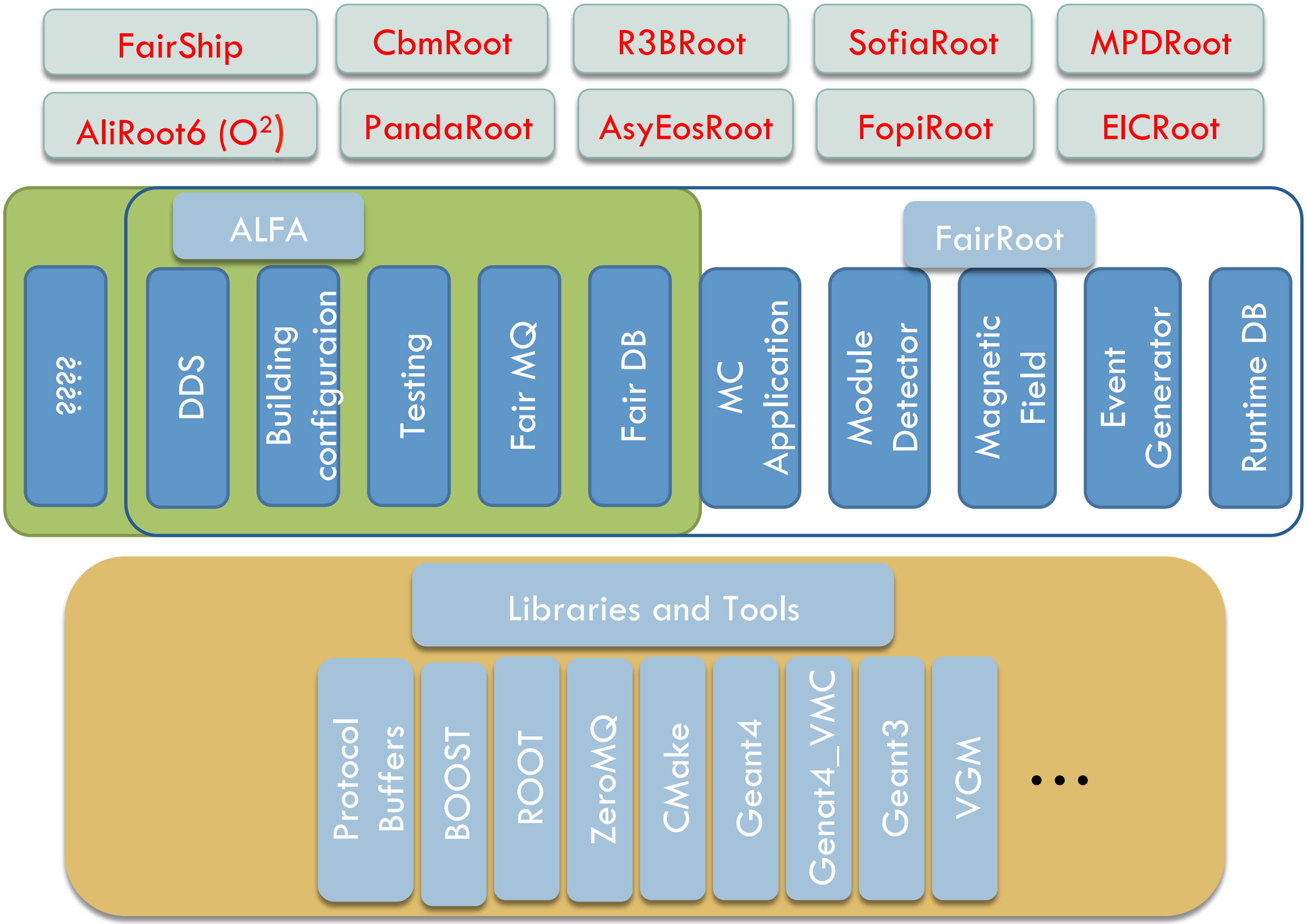


ALICE and FAIR: Why?

- A common framework will be beneficial for the FAIR experiments since it will be tested with real data and existing detectors before the start of the FAIR facility.
  - E.g.: Concepts for online calibrations and alignment can be tested in a real environment, similar to that of the planned FAIR experiments.

- ALICE will benefit from the work already performed by the FairRoot team concerning already implemented features (e.g. the continuous read-out, building and testing system, etc)

How is it with ALFA and FairRoot?



ALFA in a nutshell

- Works for online/offline as well as for simulation.
- Message queue based design.
- Modular design with user codes as plug-in
- Generic interface to transport layer
- Effective use of all available resources
- Can be deployed on a laptop, few PCs, cluster or a existing cloud system.
- Easy configuration, management and monitoring tools

ALFA/FairRoot will use Multi-processing and Multi-threading

- Multi-process concept with message queues for data exchange
  - Each "Task" is a separate process, which can be also multithreaded, and the data exchange between the different tasks is done via messages.
  - Different topologies of tasks that can be adapted to the problem itself, and the hardware capabilities.

Try to find the correct balance between reliability and performance

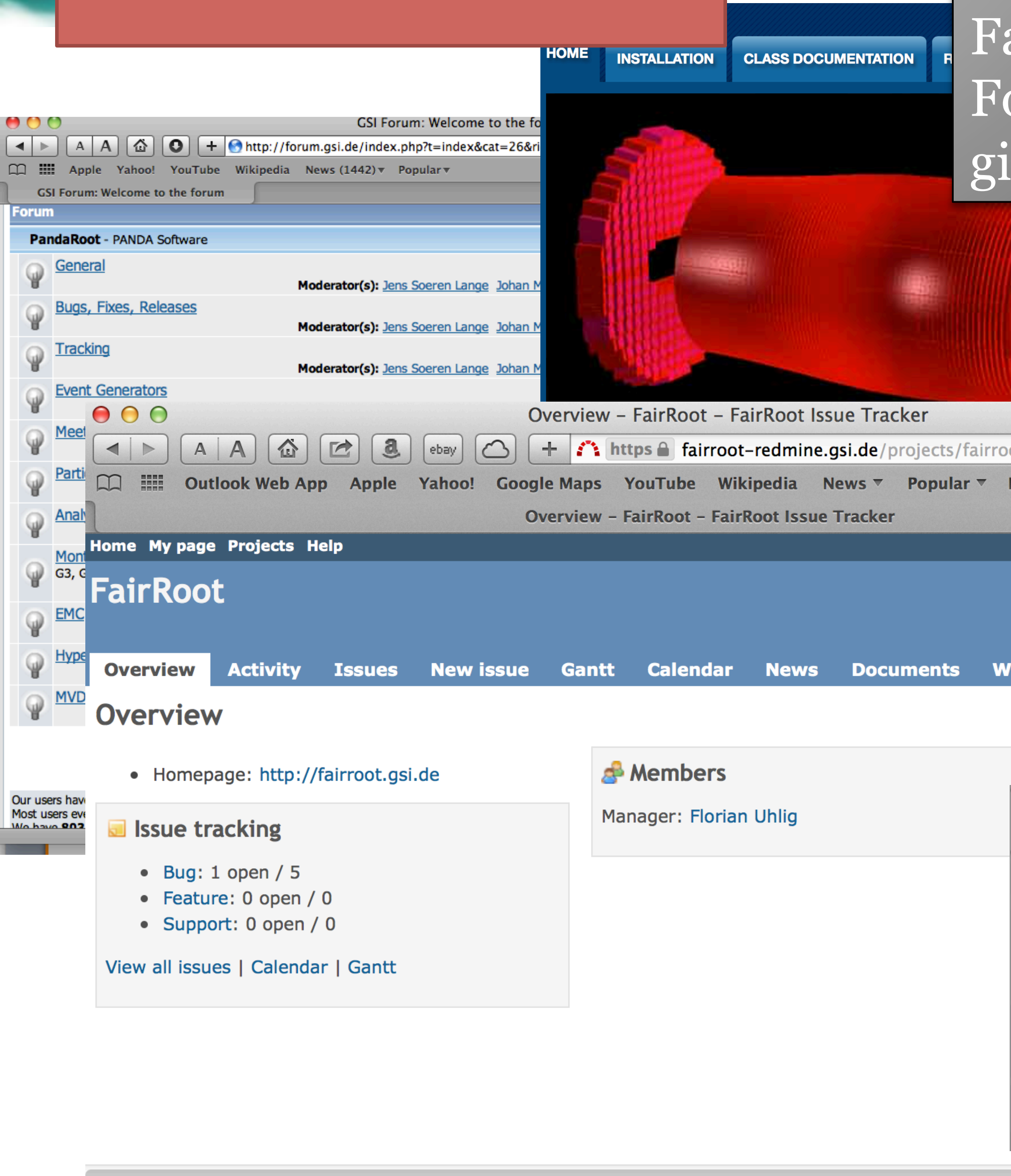
Summary

- FairRoot is meanwhile used for the whole FAIR project and many other collaborations outside FAIR
- More than 200 developer from the different experiments are contributing to the experiments code and of course from time to time to the core
- Development of general interest usually finds its way to other experiments by moving from the specific experiment implementation to FairRoot:
  - CAD TO ROOT converter
  - Event Display
  - Geane track propagator
  - Monte-Carlo validation package
  - Event generators
  - Continues read out facility
  - .....

http://fairroot.gsi.de

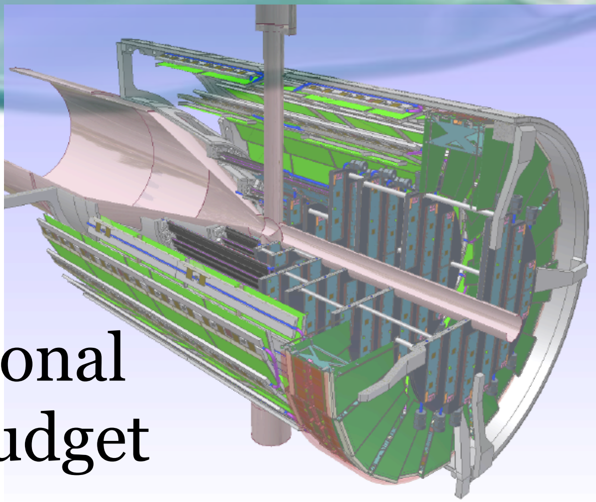
More about FairRoot

Fairroot.gsi.de
Fairroot-redmine.gsi.de
Forum.gsi.de
github.com/FairRootGroup

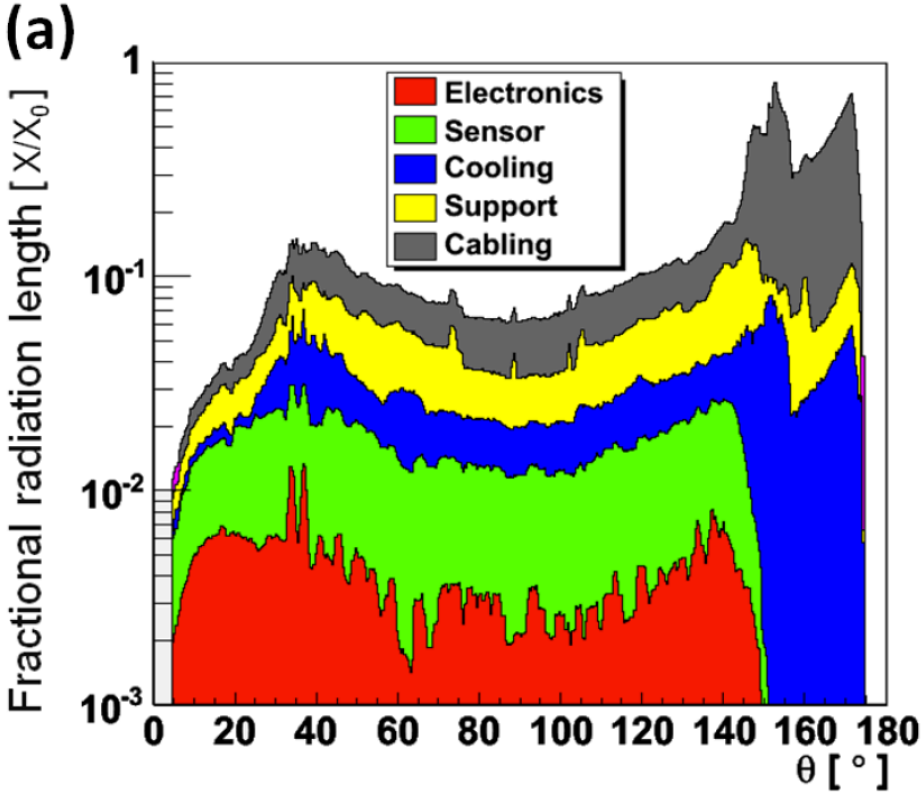


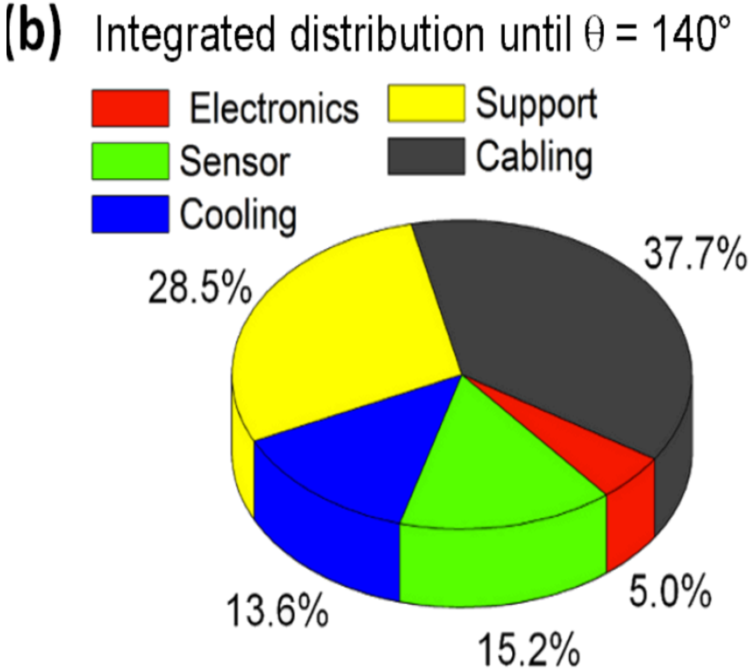
Discussion and backup

Radiation length Manager in FairRoot FairRadLenManager



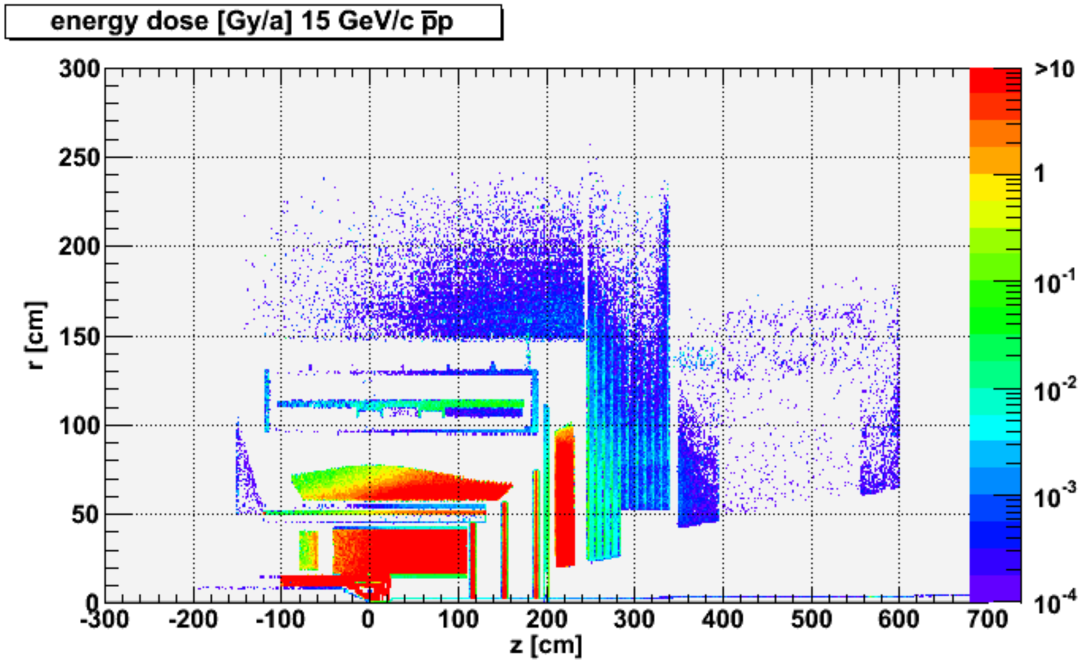
Example: Contributions of different functional parts of the MVD to the overall material budget





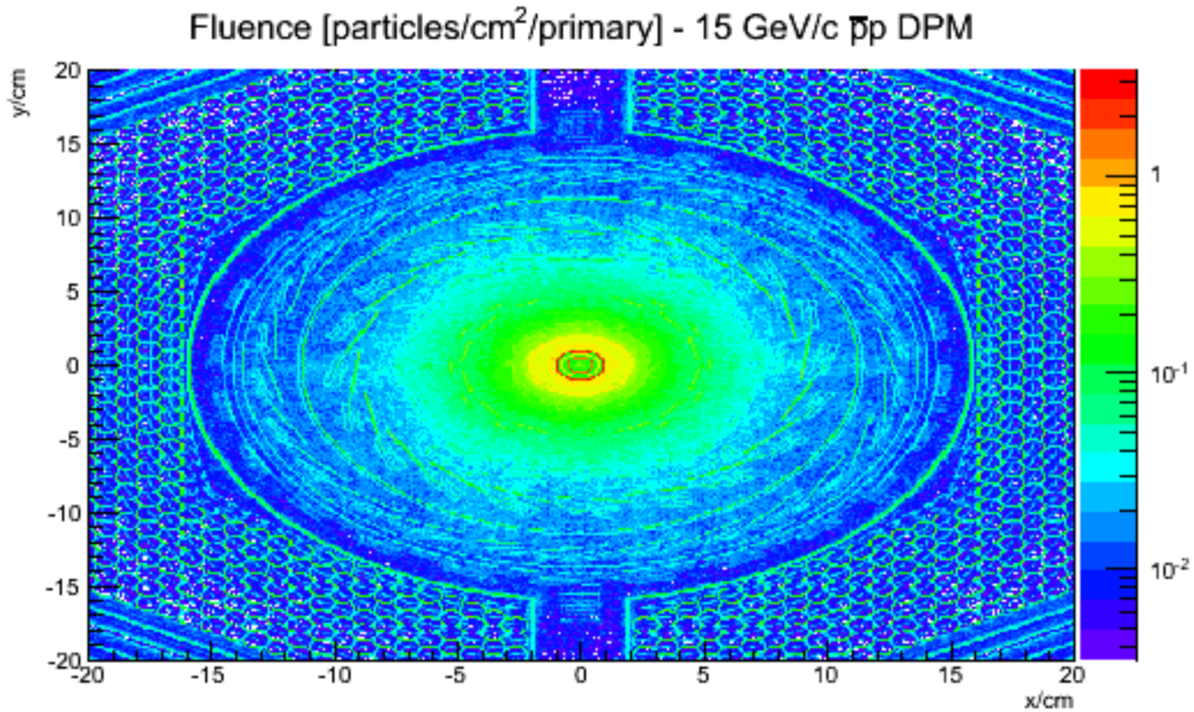
FairRadMapManager:
What energy dose will be accumulated during a certain time of operation?

- Create all physical volumes with correct material assignment
- Run the simulation engine
- FairRadMapManager will sum up every deposited energy in each volume in the geometry



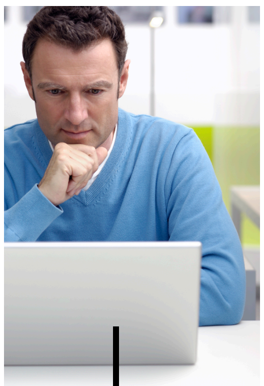
FairRadGridManager: What dose rate is expected at a certain space point/region?

Determine the particle fluence through a certain boundary (surface) and deduce a map. Knowing the volume and density of the object of interest and the specific energy loss doses can be estimated



If someone experiments with new features in his local working copy and wants to test them (experimental build)

2. Configure, build and test on local machine

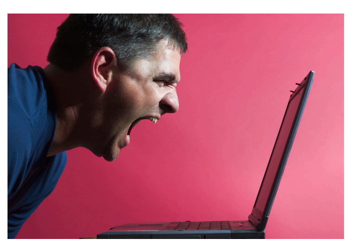


1. Update (optional)

Central SVN repository

3. Send results automatically to central web page

4. Dashboard prepares and display results



5. Developer check results

If new code enters the central code base (continuous build)

Central SVN repository

2. Basic checks:
Style, etc

Pass

Fail

Reject commit

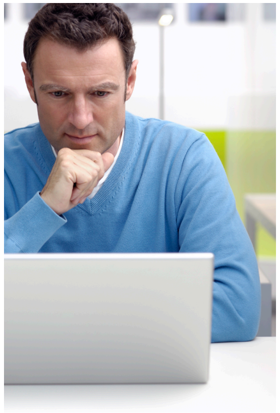
3. SVN triggers test server

Dedicated test server

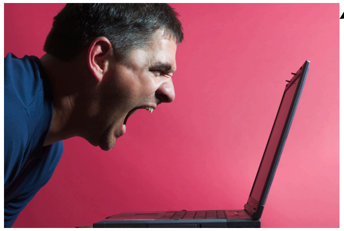
4. Configure, build and test on local machine

5. Send results automatically to central web page

1. **Developer commit code**



7. In case of problems Dashboard sends an E-mail to Developer and Administrator



Login | Dashboards — Wednesday, March 17 2010 12:36:45 CET

CBMROOT Dashboard

DASHBOARD CALENDAR PREVIOUS CURRENT PROJECT

No update data as of 2010-03-16T23:00:00 CET — Help

[Show Filters]

Nightly

Site	Build Name	Update		Configure			Build			Test				Build Time	Labels
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min		
depc165.gsi.de	FC8-linux-i686-gcc4.1.2-fairsoft_jul09	24	0.2	0	0	0	0	50 +49	24.1	0	1 +1	7 +7	17.3	2010-03-17T00:20:28 CET	(none)
fwklux2.fz-rossendorf.de	lenny-sid-GNU_Linux-i686-gcc4.3.2-fairsoft_jan10	21	0.7	0	0	0	0	50 +49	5.3	0	1 +1	7 +7	9.1	2010-03-17T05:01:17 CET	(none)
fwklux5.fz-rossendorf.de	lenny-sid-GNU_Linux-x86_64-gcc4.3.3-fairsoft_jan10	21	0.6	0	0	0.1	0	50 +49	2.2	0	1 +1	7 +7	6.2	2010-03-17T01:01:11 CET	(none)
demac006	MacOSX10.5-darwin-i386-gcc4.2.1-fairsoft_jul09	21	0.1	0	0	0.1	0	50 +49	2.2	0	1 +1	7 +7	5.3	2010-03-17T00:30:24 CET	(none)
h035.nuph.us.edu.pl	openSUSE-10.3-linux-i686-gcc4.2.1-fairsoft_jul09	21	1	0	0	0.7	0	50 +49	22.2	0	1 +1	7 +7	13	2010-03-17T01:32:41 CET	(none)
Totals	5 Builds	108	2.6	0	0	0.9	0	250	56	0	5	35	50.9		

No Nightly [Mar08] Builds

Continuous

Site	Build Name	Update		Configure			Build			Test				Build Time	Labels
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min		
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	1	0.2	0	0	0	0	0	0	0	1	7	5.9	2010-03-17T10:47:00 CET	(none)
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	2	0.1	0	0	0	0	0	0	0	1	7	5.9	2010-03-17T10:04:29 CET	(none)
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	3	0.2	0	0	0.2	0	0	0.1	0	1	7	6.4	2010-03-17T09:57:41 CET	(none)
Totals	3 Builds	6	0.5	0	0	0.2	0	0	0.1	0	3	21	18.2		

No Experimental Builds

No Coverage

No Dynamic Analysis

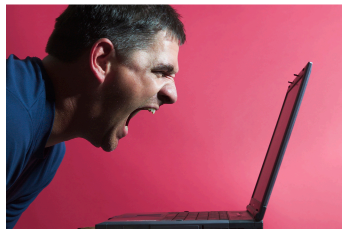
Kitware — CDash 1.4.1 © 2009 Kitware Inc. [report problems]

6. Dashboard prepares and display results

From time to time a full check on all supported platforms should be done (nightly build)

Central repository

1. Update



4. In case of problems Dashboard sends an E-mail to Developer and Administrator

5. In the morning Developers and Administrators check their mails and the dashboard. And the development cycle starts again

2. Send results automatically to central web page

3. Dashboard prepares and display results

Login | Dashboards   Wednesday, March 17 2010 12:36:45 CET

CBMROOT Dashboard

DASHBOARD CALENDAR PREVIOUS CURRENT PROJECT

No update data as of 2010-03-16T23:00:00 CET   Help

[Show Filters]

Nightly

Site	Build Name	Update		Configure			Build			Test				Build Time	Labels
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min		
depc165.gsi.de	FC8-linux-i686-gcc4.1.2-fairsoft_jul09	24	0.2	0	0	0	0	50 +49	24.1	0	1 +1	7 +7	17.3	2010-03-17T00:20:28 CET	(none)
fwklux2.fz-rossendorf.de	lenny-sid-GNU_Linux-i686-gcc4.3.2-fairsoft_jan10	21	0.7	0	0	0	0	50 +49	5.3	0	1 +1	7 +7	9.1	2010-03-17T05:01:17 CET	(none)
fwklux5.fz-rossendorf.de	lenny-sid-GNU_Linux-x86_64-gcc4.3.3-fairsoft_jan10	21	0.6	0	0	0.1	0	50 +49	2.2	0	1 +1	7 +7	6.2	2010-03-17T01:01:11 CET	(none)
demac006	MacOSX10.5-darwin-i386-gcc4.2.1-fairsoft_jul09	21	0.1	0	0	0.1	0	50 +49	2.2	0	1 +1	7 +7	5.3	2010-03-17T00:30:24 CET	(none)
h035.nuph.us.edu.pl	openSUSE-10.3-linux-i686-gcc4.2.1-fairsoft_jul09	21	1	0	0	0.7	0	50 +49	22.2	0	1 +1	7 +7	13	2010-03-17T01:32:41 CET	(none)
Totals	5 Builds	108	2.6	0	0	0.9	0	250	56	0	5	35	50.9		

No Nightly [Mar08] Builds

Continuous

Site	Build Name	Update		Configure			Build			Test				Build Time	Labels
		Files	Min	Error	Warn	Min	Error	Warn	Min	NotRun	Fail	Pass	Min		
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	1	0.2	0	0	0	0	0	0	0	1	7	5.9	2010-03-17T10:47:00 CET	(none)
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	2	0.1	0	0	0	0	0	0	0	1	7	5.9	2010-03-17T10:04:29 CET	(none)
lxir010.gsi.de	Etch64-GNU_Linux-x86_64-gcc4.1.2-fairsoft_jan10	3	0.2	0	0	0.2	0	0	0.1	0	1	7	6.4	2010-03-17T09:57:41 CET	(none)
Totals	3 Builds	6	0.5	0	0	0.2	0	0	0.1	0	3	21	18.2		

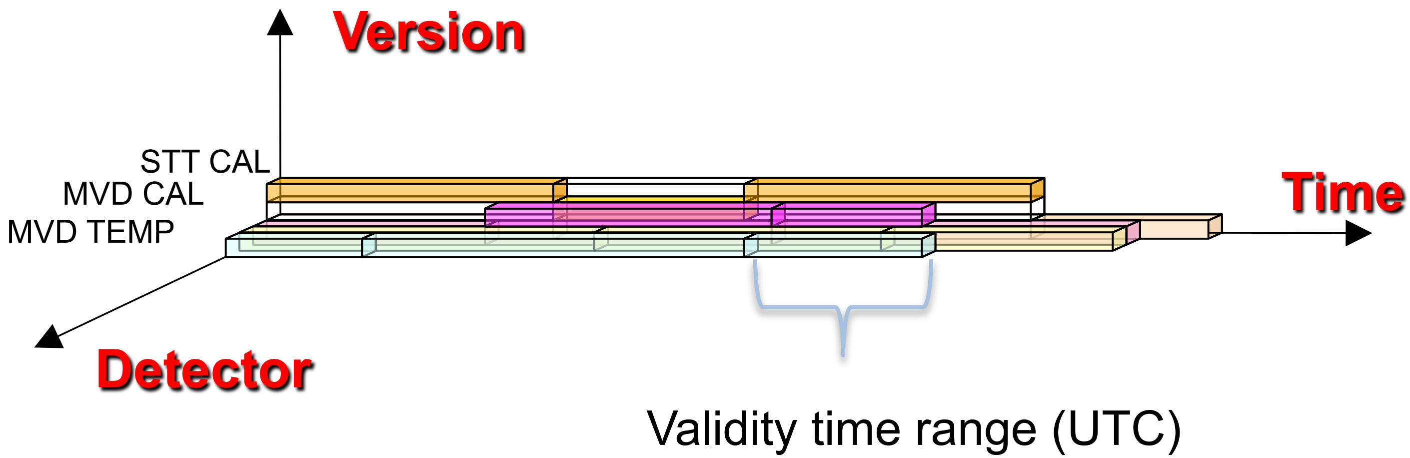
No Experimental Builds

No Coverage

No Dynamic Analysis

Kitware   CDash 1.4.1 © 2009 Kitware Inc. [report problems]

Version Management



The Query process

1. Context (Timestamp,Detector,Version) is the primary key
2. Context converted to unique SeqNo
3. SeqNo used as keys to access all rows in main table
4. System gives user access of all such rows

Integrating the existing software: