Reconstruction Software Status

A. Sarti

Introduction

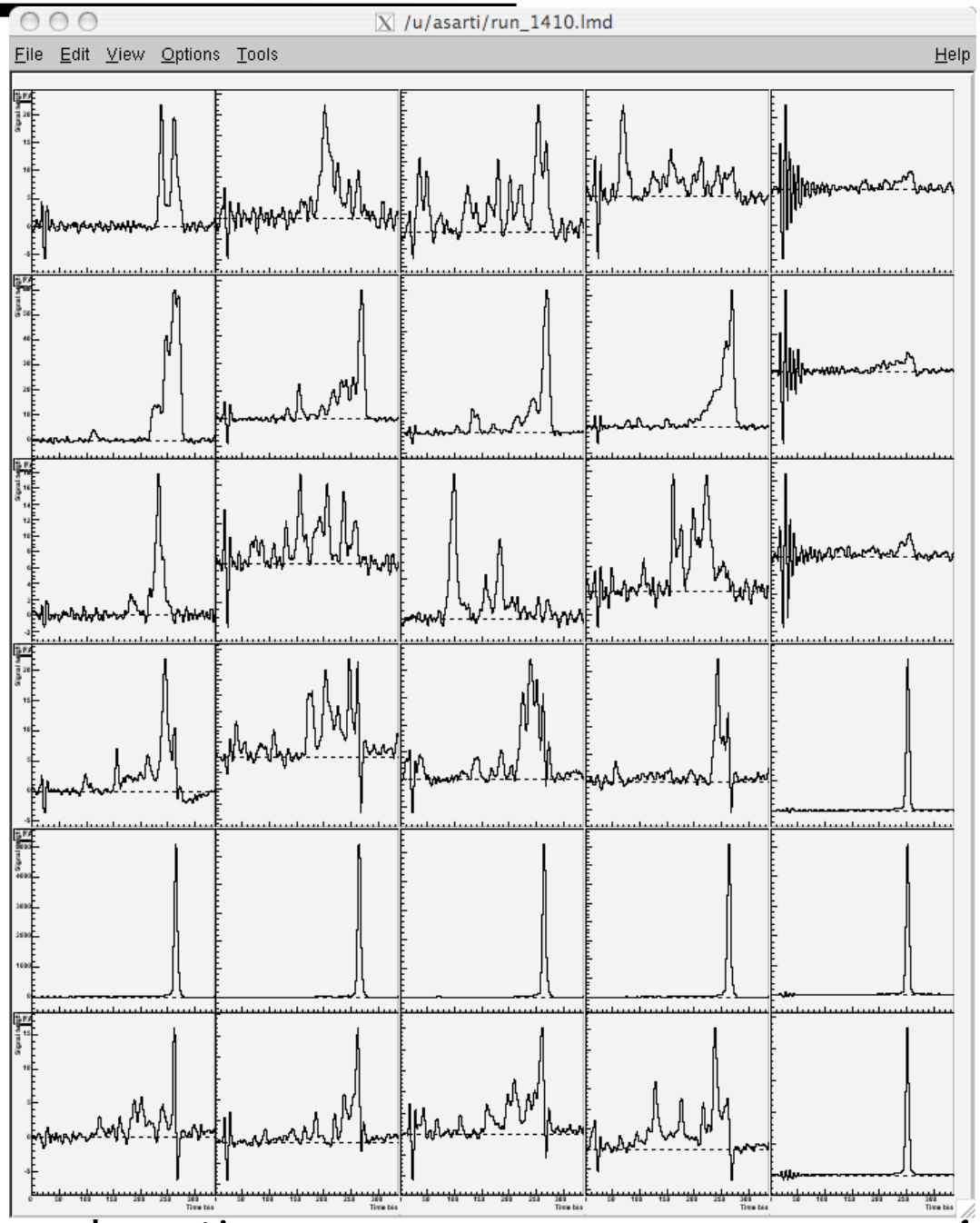
- Project up and running!
  - mailing list: first-soft@gsi.de has been setup. if you know about anyone interested in contributing let me know his name and I'll add it to the mlist.
  - twiki pages: http://wiki.gsi.de/cgi-bin/view/FIRST/ReconstructionSoftware
    - Pages documenting the simulation and reconstruction status have been created. You'll find there all the informations needed to get the code, compile it and contribute to its development
  - svn repository: (svn → tool to work with shared code)
    - Currently all the code lives under an SVN repository on a machine @ Uniroma 1 (La Sapienza). I hope that soon i'll be able to migrate it to the gsi repository [made request >3 week ago!!!]
  - Meetings
    - Held and EVO meeting the past 10th march
    - Indico meetings can be found here

DAQ

- What has been used so far (and believed to be working still :)...)
  - f_user + W. Mueller classes: data is written by MBS into the lmd file and the WM classes are already configured to deal with Camac, Fastbus and VME boards.
  - Currently working (me + V. Monaco) on the implementation of new detectors in that framework: **we will be able to use all the code already developed by Music and ToF wall for calibration + online monitoring.**
- Things I know:
  - ToF wall is happy with f_user + WM approach: they are currently using it to check the ToFW status after/during its 'rebuilding'.
  - The latest Music acquisition (pulser run) was done using the f_user/MBS setup and data was displayed with the 'display QAD' routines... Next important step, after dealing with 'gas' issues: acquire a cosmics run and process it....

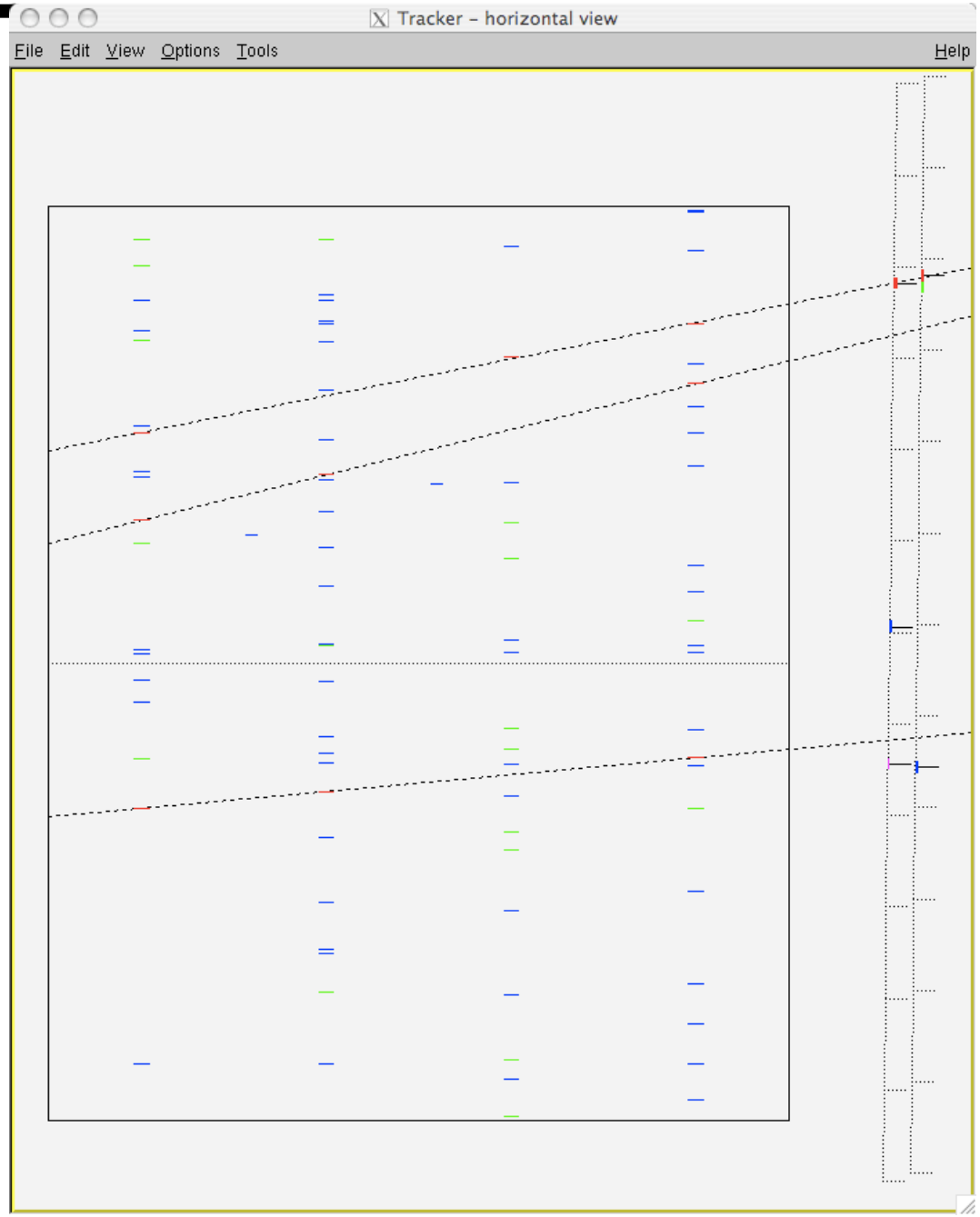
Online monitoring

```
    TAGmbsEventFilterType* p_ef_fbus =
new TAGmbsEventFilterType();
    p_ef_fbus->AddType(32,1000,1999);
    TAGmbsEventFilterType* p_ef_land =
new TAGmbsEventFilterType();
    p_ef_land->AddType(32,3000,3999);

    TAGactMbsReader* a_inl = new
TAGactMbsReader("a_in");

    a_inl->SetupChannel(r_camac, new
TAGmbsEventFilterType(34));
    a_inl->SetupChannel(r_fbus, p_ef_fbus);
    a_inl->SetupChannel(r_land, p_ef_land);
    a_inl->SetupChannel(r_fadc,  new
TAGmbsEventFilterType(40));
```



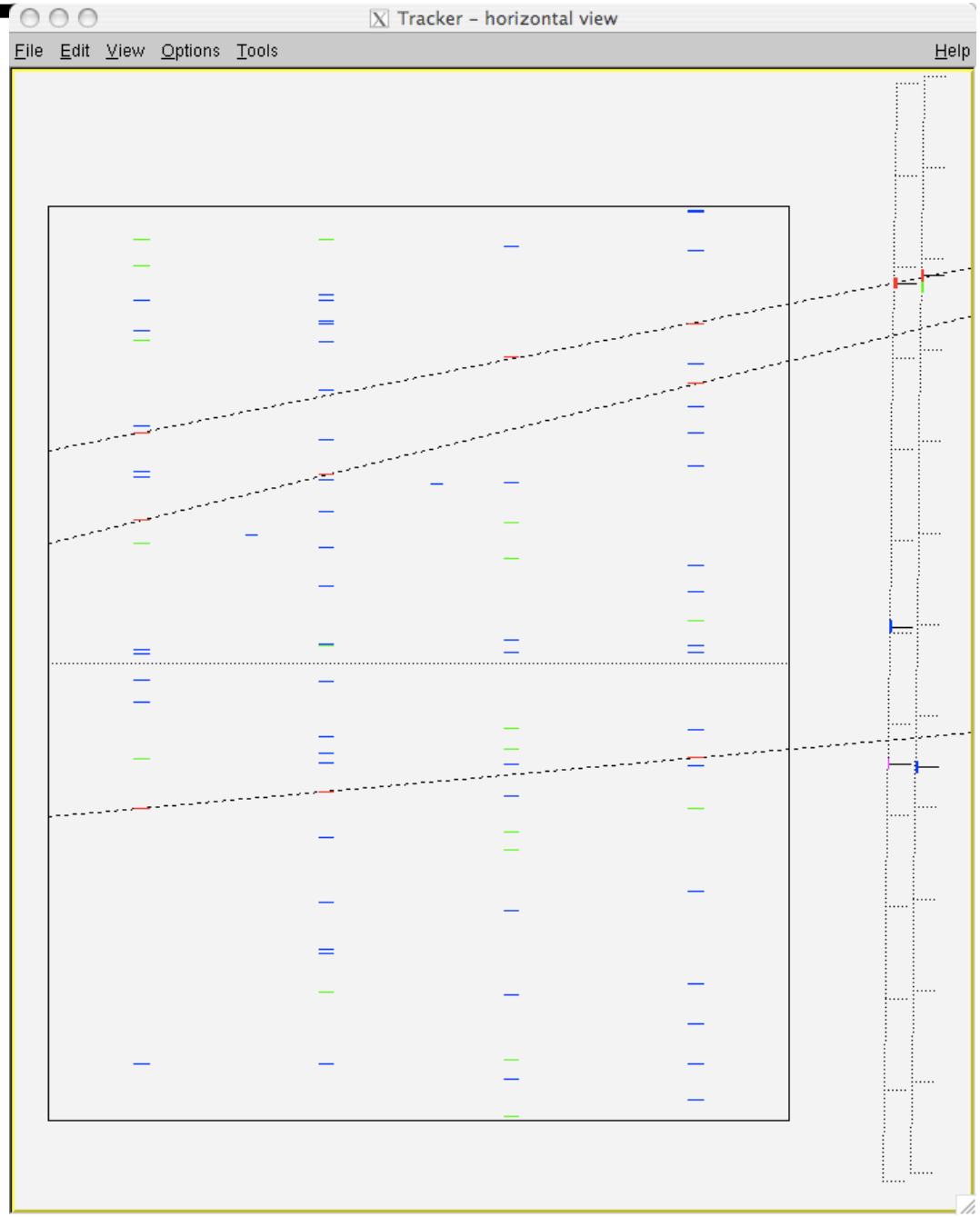
Low level DATA decoding/monitoring

- We
  - Are in a good shape for decoding and monitoring what's happening to the Music and TofWall.
  - Need to implement ALL the new detectors reading trough the WM framework
- For the new detectors
  - Templates based on where we're going to get the info from (VME?, FastBus? etc etc) are being prepared [will become crucial once we get to the point of making a test beam read by mbs]



Low level DATA decoding/monitoring (II)

- Software being assembled under l0reco directory in svn repository
- Not yet available since it works only at GSI and, for now, it will be used only by ToF and Music 'newbies' trying to understand what they're really doing.
- If you have any urgence and you have a GSI account we can go trough the necessary steps needed to obtain the plots on the right!



Low level MC decoding/monitoring

- Crucial to
  - understand the goodness of our beautiful MC
  - train people to work with WM classes
  - Have a 'FIRST' event that can be fed to the global reconstruction and train the reconstruction algorithms.
- Difficult (for the Music) because:
  - You need to have a 'somehow' deep knowledge of the detector in order to understand how to match the info from the MC and the info from the Detector
- Easier (for all the rest) because:
  - Code written from scratch anyhow :) + there is a very good matching with what comes from MC and what are already using in our 'local' reconstruction.

So far our largest effort was put on starting the Music and ToF wall MC interface implementation

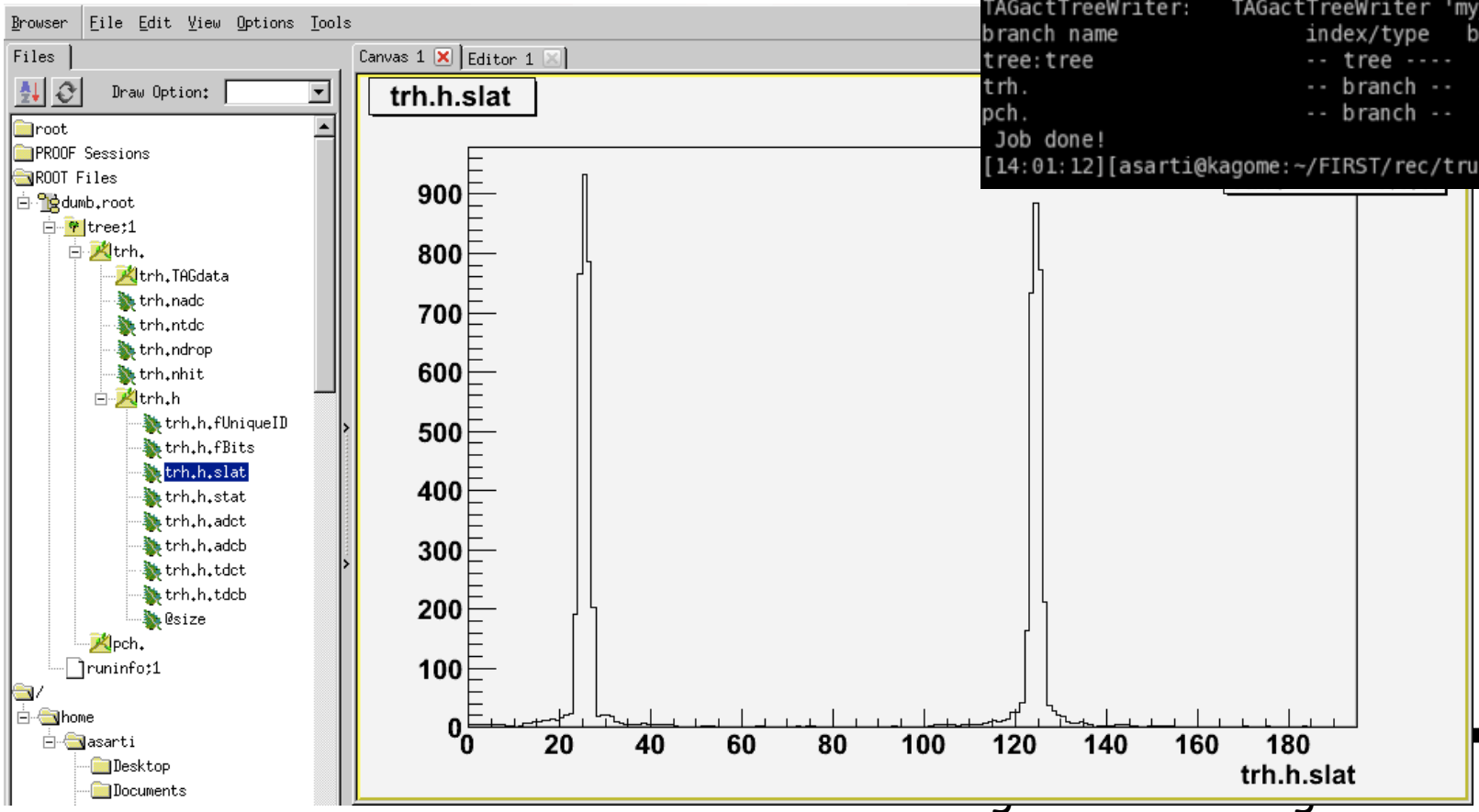
Example : TofWall

- The development of MC-reconstruction interface for theTof Wall raw information was the first exercise performed.
- Together with Till we managed to
  - Develop the classes that are interfacing the MC structure (event) with the TofWall WM ntuplizer.
  - Use the WM framework (tagroot writing) to fill a root file with MC info and WM structure that can be
- To run the code, follow the instructions on the Twiki page.

ToF Wall: it works!

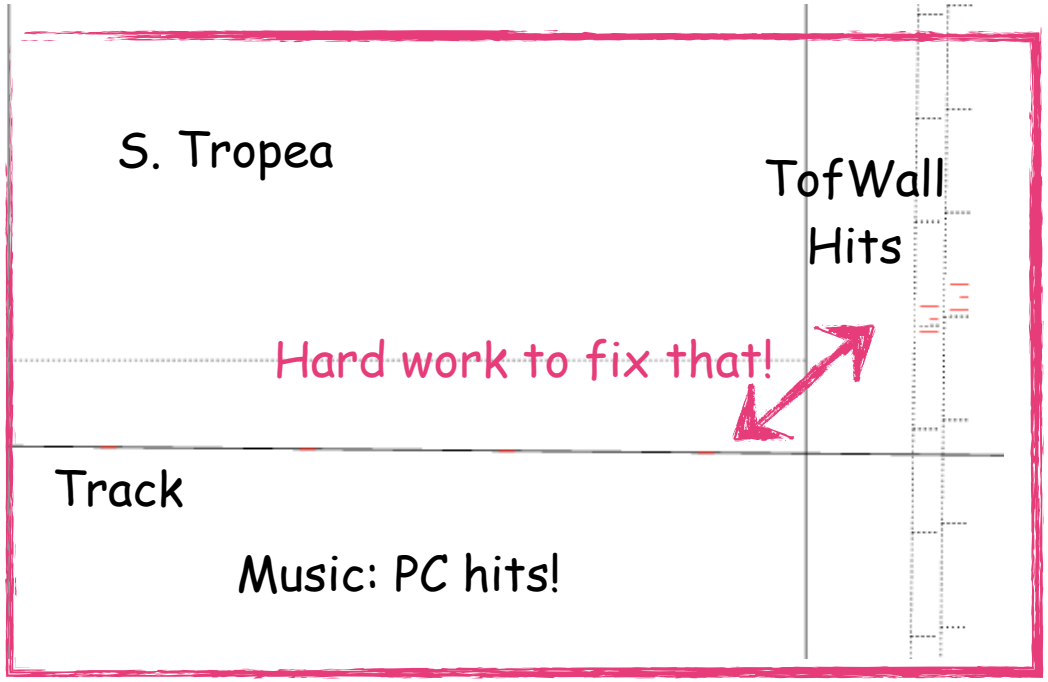
- Takes as input the mc in
  - data/mc/MC_Evt1k_test.root
  - dumps the hits info in the dumb.root output

```
[14:00:41][asarti@kagome:~/FIRST/rec/trunk/l0mcreco]$./DecodeMC
Opening ROOT outp file:: dumb.root ../data/test.txt
 Job started!
TFile**         ../data/mc/MC_Evt1k_test.root
 TFile*         ../data/mc/MC_Evt1k_test.root
  KEY: TTree    EventTree;1     ions
Loading p_pcmap for cam/run = -1/-1
Loading p_pcgeo for cam/run = -1/-1
Loading p_pchit for cam/run = -1/-1
-1 -1
   drift setup: 'p'  fac: 0.000   ref:  12.5
 Beginning the Event Loop
Running against ntuple with:: 1000 entries!
 Loaded Event:: 0
 Loaded Event:: 100
 Loaded Event:: 200
 Loaded Event:: 300
 Loaded Event:: 400
 Loaded Event:: 500
 Loaded Event:: 600
 Loaded Event:: 700
 Loaded Event:: 800
 Loaded Event:: 900
 Ending the Event Loop
TAGactTreeWriter:   TAGactTreeWriter 'my_out'
branch name              index/type   bt bid  tot size file size   comp   frac
tree:tree                -- tree ----          3372.3k    117.1k  28.81
trh.                     -- branch --  0  -2    191.7k    134.6k   1.42 115.0%
pch.                     -- branch --  0  -2   3567.9k    194.6k  18.34 166.2%
 Job done!
[14:01:12][asarti@kagome:~/FIRST/rec/trunk/l0mcreco]$
```

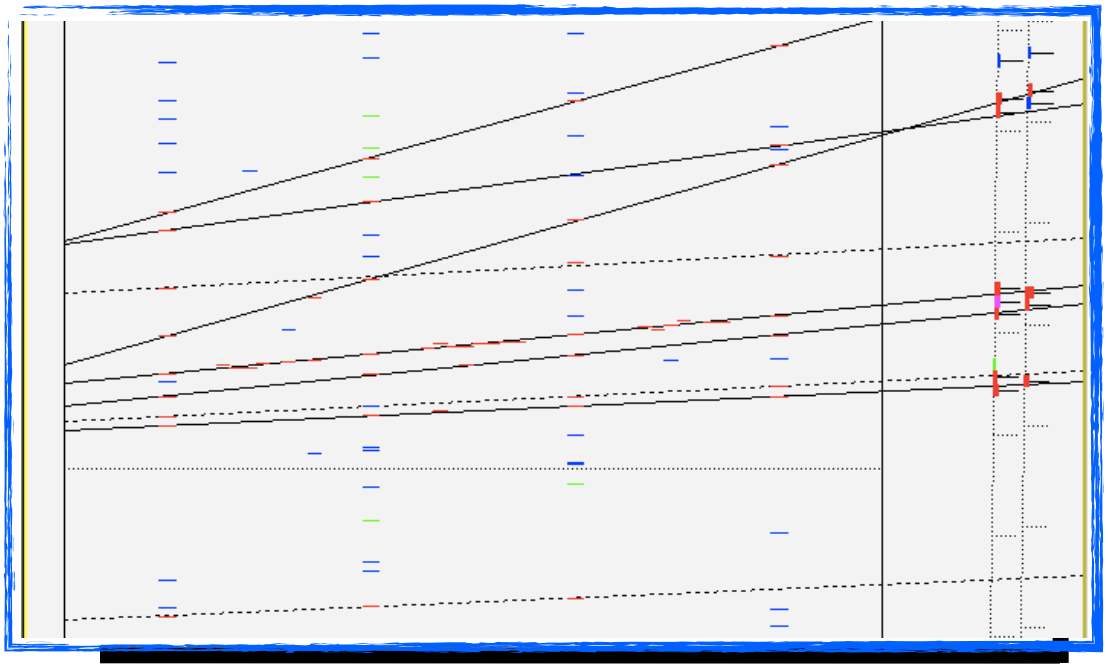


Increased difficulty: Music!

- Classes that handle the Music info are much more 'structured' and require a deeper data/code/hw understanding
  - We started one week ago to implement the interfaces...
  - Really hard/useful: necessary to match the Geometry conventions of data and MC, understand data structure, data processing..... And also check/adapt the initialization of 'detector parameters' (drift velocity, scale factors, etc etc) tuned on MC to match data.



Going from **THAT** to..... **that**: lot of work on... Geometry, Parameters.



Next steps

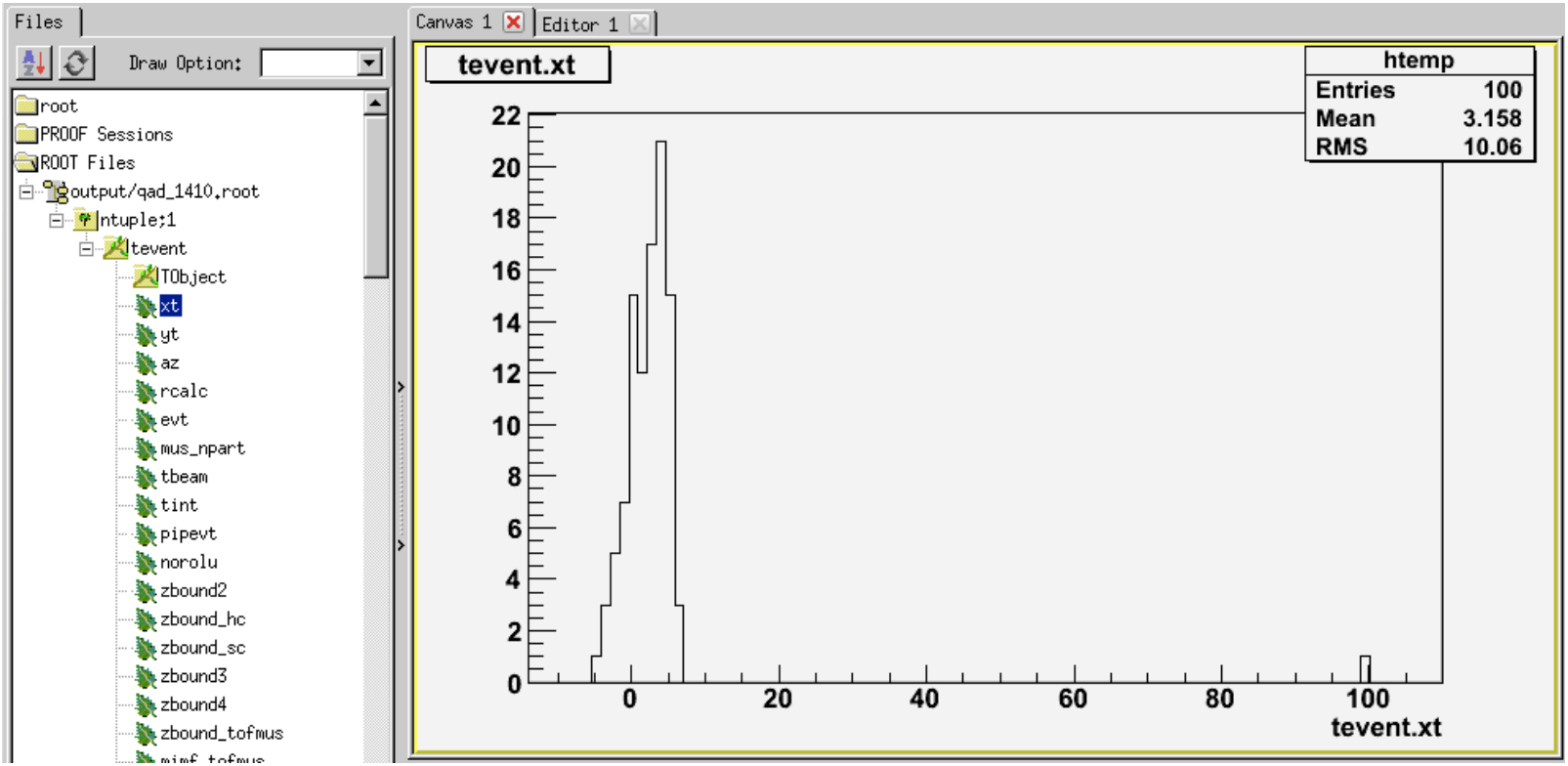
- ToF wall example can be used by others to start the development of their own classes
- Subdetector status:
  - ToF: raw hit dumping implemented, should work on the geometry and on the configuration handling
  - Music: the interfaces to Music PC hit and trk classes are under development (S. Tropea & others). First 'plots' of raw quantities and event displays are available (see slide before)
  - IR detectors: we are starting from scratch. I'll provide soon a template for the NEW detectors (since not only the action but also the data description is lacking)
- Framework: the implementation of few 'common' methods to configure the job, handle the geometry, the input/output, etc etc **is ongoing**:
  - If you need something: do not sit and wait... just shout and I'll try to take care of the technical problem.

High level reconstruction

- Taken from Titti work area:
  - it is the only working example of an 'high level reconstruction' that process the ToF and Music data together and fills some 'event' and 'particle' classes
- Lives under hlreco in svn
  - Takes as input the output of the previous step (root file written using tagroot)
- To compile it and run it.... go on the twiki site for the instructions!
  - DISCLAIMER: this code is JUST a template to be used by people intrested in developing high level reconstruciton methods.....
  - Cleanup, rationalization, reorganization of that code is foreseen and necessary in the future (as soon as we will start to work on track matching on MC events!!!)

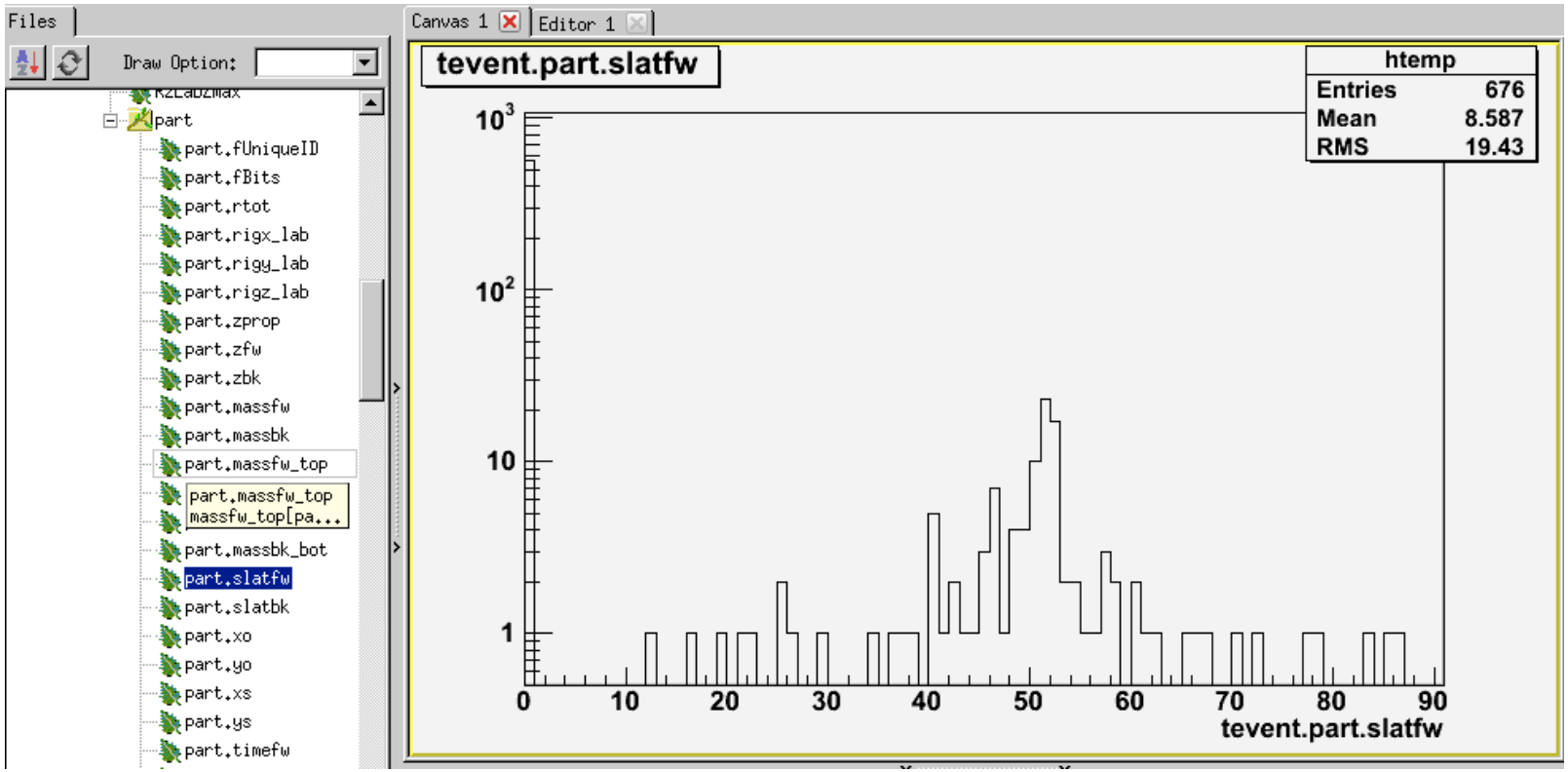
HlReco: it works

- The existing code has already been tested on old decoded data
  - A major step would be to check that code against the MC… But for that we need first to complete the steps in the previous pages of this talk



HlReco: it works (II)

- Particle class available as well
- Everything obtaind from the processing of an already decoded 2006 data file, using HlReco exec.



Next steps

- As soon as we have a full FIRST decoded event, the full reconstruction will be also testable on the MC
  - Of course, in the meanwhile we will start to match 'separately' Tof, Music and BM or BM and Vertex, etc etc
- People can already look at the reconstruction workflow, understand the code, find out the places where we need to adapt the logic since we're going to have other infos from other detectors.
- The code also needs to be:
  - cleaned up
  - improved to handle the calibrations input
  - checked against latest reconstruction versions [full / detailed xcheck with existing codes] and possible bugs introduced by me :) ....

Conclusions

- Basic building blocks have been collected, assembled and tested
  - Raw data decoding: works @ GSI, still using Titti macros to produce plots, will be organized and released under svn soon. **TO DO**: **implement data description and Actions** to fill data from mbs (coordination with V. Monaco, mandatory!) for the IR detectors
  - MC decoding: templates for ToF and Music decoding are released and available. Starting to look at distributions: is really helpful in order to improve out MC and understand our detectors!!!! **TO DO**: **same as previous point for IR detectors** [but easy, since MC simulation was 'driven by' requests of det. experts and we start from scratch!], **Improve the ToF and Music classes**... that will be hard.. BUT extremely useful!!
  - High Level Reconstruction: an executable that performs the backtracking is released under SVN. **TO DO**: clean up and recheck the current code. Then **implement the handling of IR information and track matching**. The aim is to have a 'full reconstructed FIRST' event display.