Status of ILD software

Towards a simulation baseline

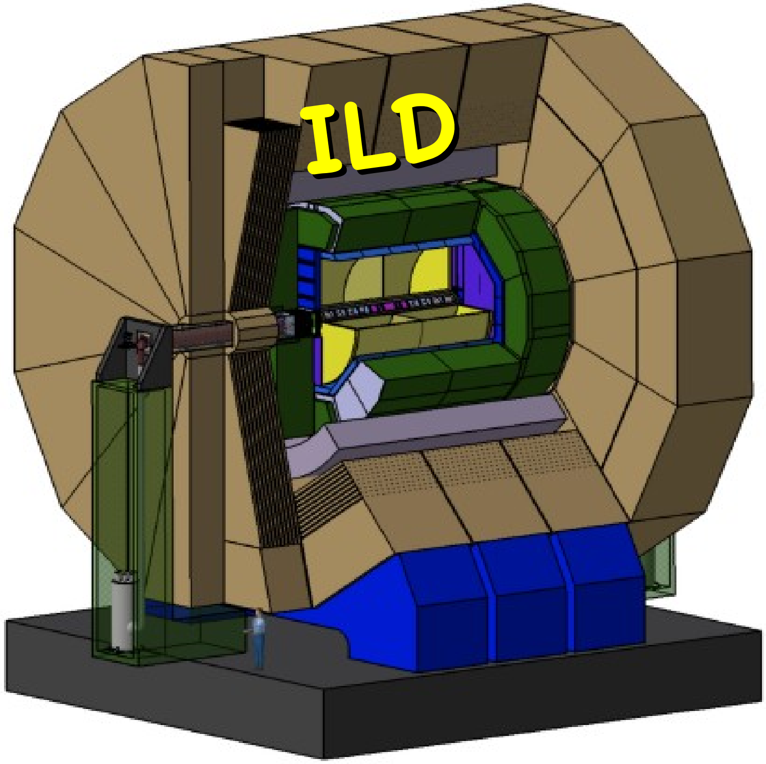
Frank Gaede, DESY

International Workshop on Linear Colliders

CERN, Oct. 18-22, 2010

Outline

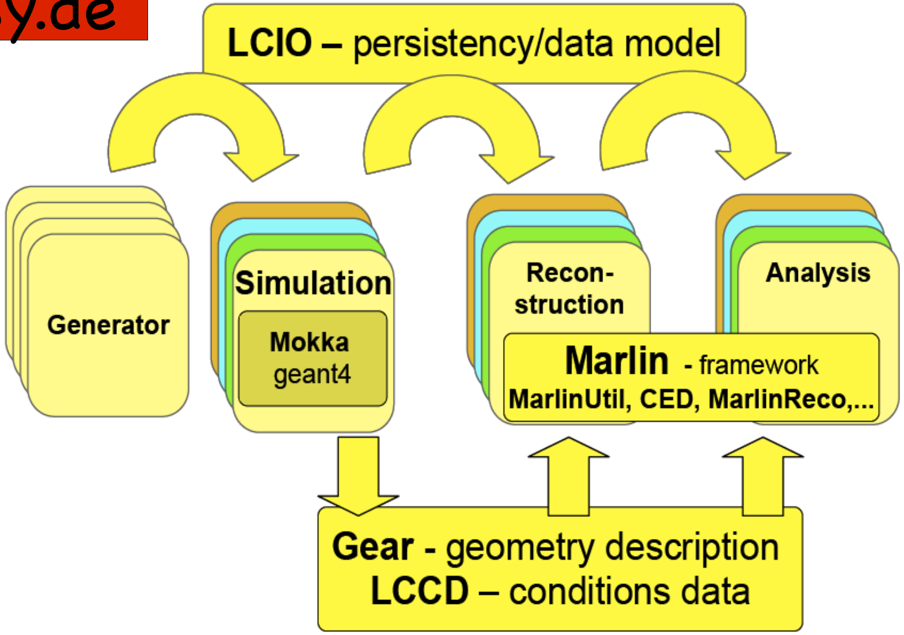
- Introduction & Overview
- new developments since LCWS2010
  - core software
  - reconstruction
  - Mokka
- towards a simulation baseline
  - status - next steps
  - discussion



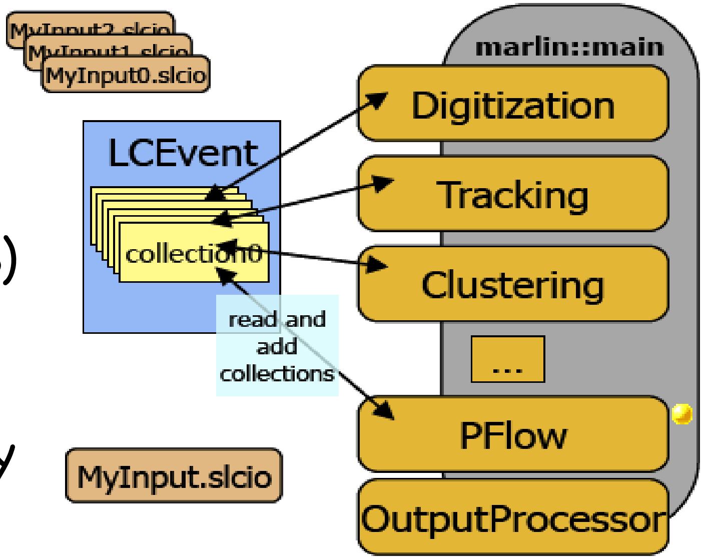
ILD Core Software Tools

http://ilcsoft.desy.de

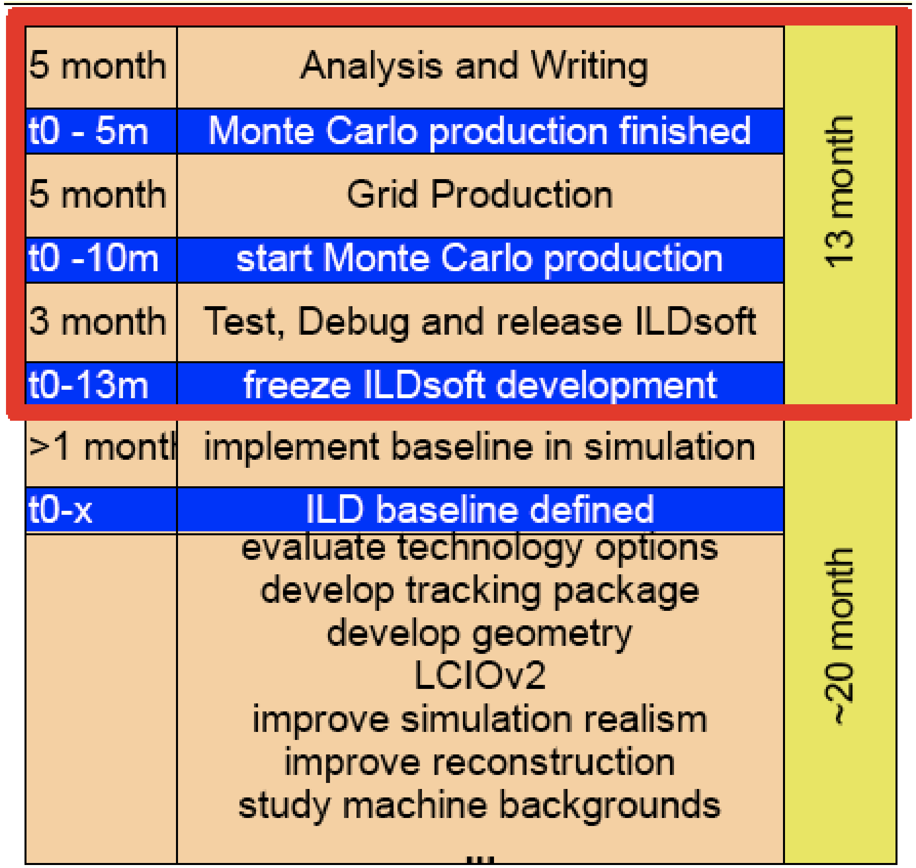
- **Mokka** (LLR)
  - geant4 simulation application
- **LCIO** (DESY/SLAC)
  - international standard for persistency format / event data model
- **Marlin**
  - core application framework for reconstruction & data analysis
- **GEAR** geometry package f. reconstruction
- **LCCD**
  - conditions
  - data toolkit (DB)
- **CED**
  - 3d event display



- complete framework used in Monte Carlo & 'real experiments':
  - **ILD detector concept** studies
  - **Calice** calo testbeam
  - **LC-TPC** testbeam
  - EUDET - **Pixel Telescope**
- **synergies between testbeam and global detector optimization**

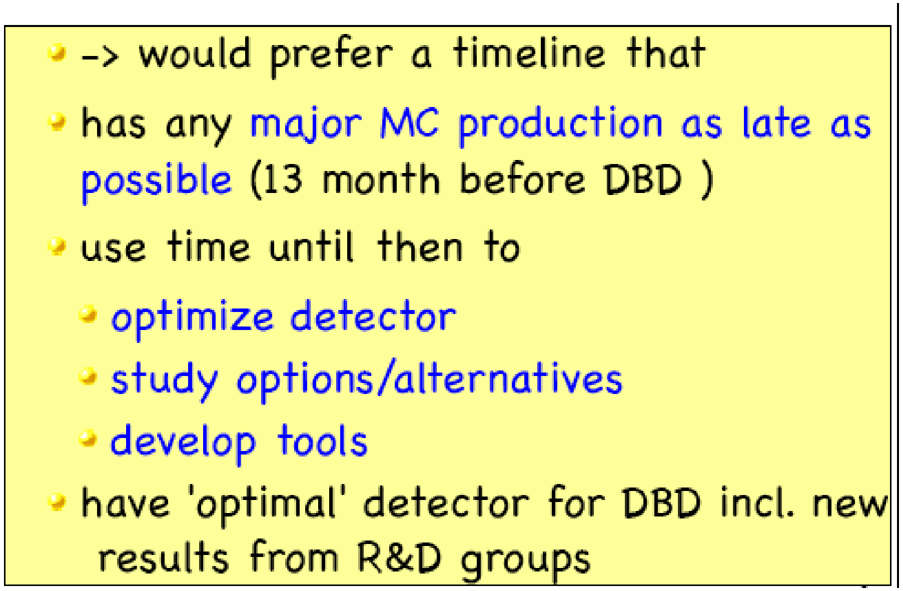


software timeline



5 month	Analysis and Writing	13 month
t0 - 5m	Monte Carlo production finished	
5 month	Grid Production	
t0 -10m	start Monte Carlo production	
3 month	Test, Debug and release ILDsoft	
t0-13m	freeze ILDsoft development	
>1 montl	implement baseline in simulation	~20 month
t0-x	ILD baseline defined	
	evaluate technology options develop tracking package develop geometry LCIOv2 improve simulation realism improve reconstruction study machine backgrounds ...	

at Beijing ILD meeting proposed:



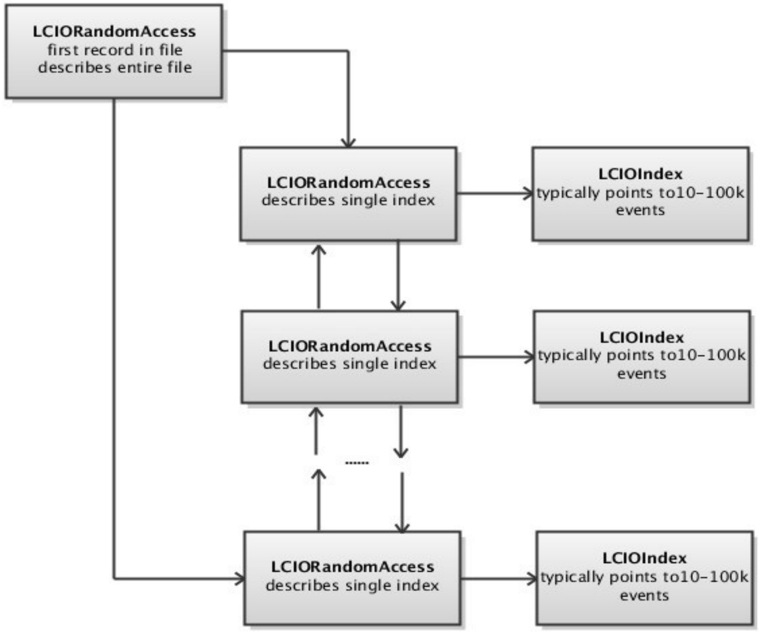
- -> would prefer a timeline that
- has any major MC production as late as possible (13 month before DBD)
- use time until then to
  - optimize detector
  - study options/alternatives
  - develop tools
- have 'optimal' detector for DBD incl. new results from R&D groups

- this schedule has been accepted
- many developments since then
  (partly presented at
  Software & Integration WS
  at DESY)

2010-2011: time to develop new reconstruction code and improve core software tools

towards LCIOv2 - v01-51-01

- LCIO provides a rather complete event data model and has been used successfully in SID and ILD LOI mass production and in various R&D test beam programs
- LCIOv2 needs to be backward compatible and should provide some new features
  - **direct access to events -> DONE**
  - **partial reading of events**
  - **splitting of events over files**
  - **(storing of arbitrary user classes)**
  - **simplify using LCIO with ROOT -> DONE**
    - (ROOT macros, TTreeViewer, I/O (?) ,...)
  - **improving the event data model -> started**
    - (1d,2d hits, tracks/trajectories)

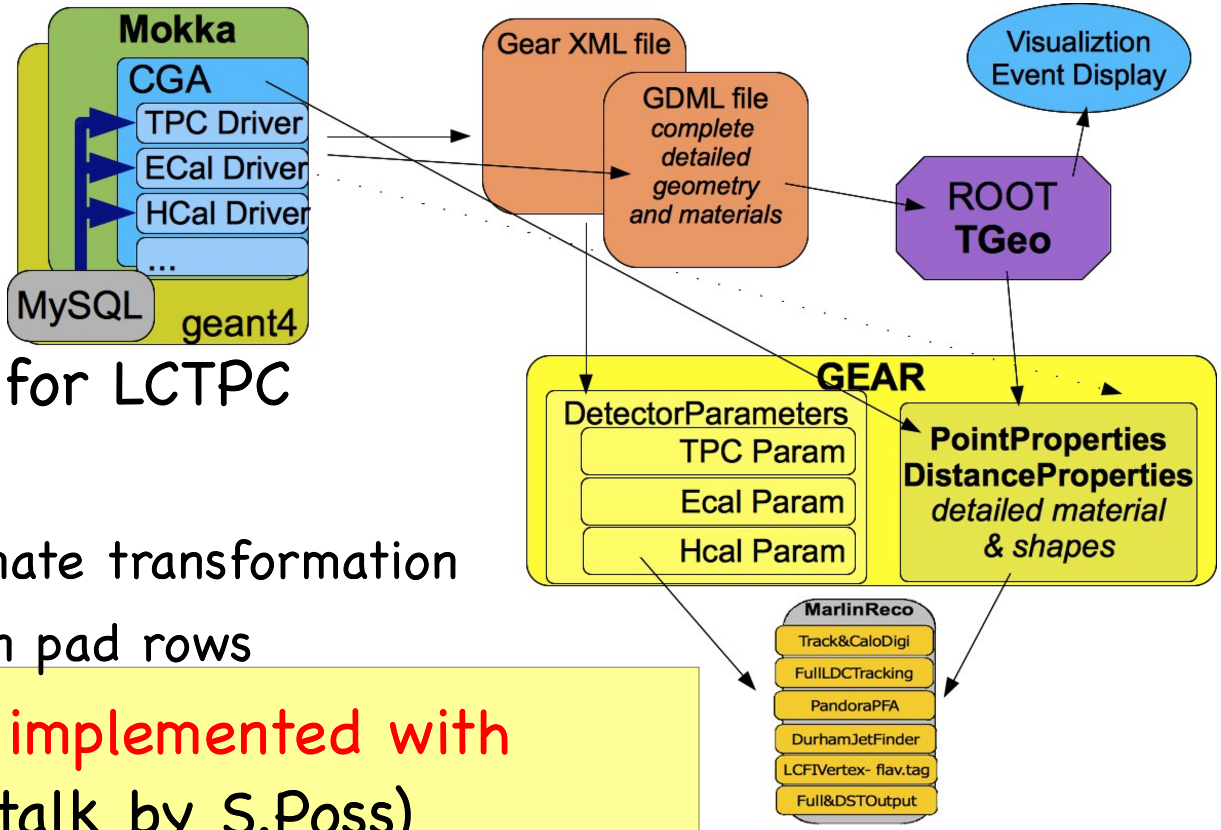


direct Access:

- record written at close()
- can append to files
- can add to existing OLD files (if opened in write mode)

- new ostream operators<<(...) in C++
- cout << ((MCParticle*) c->getElementAt(i)) << endl ;

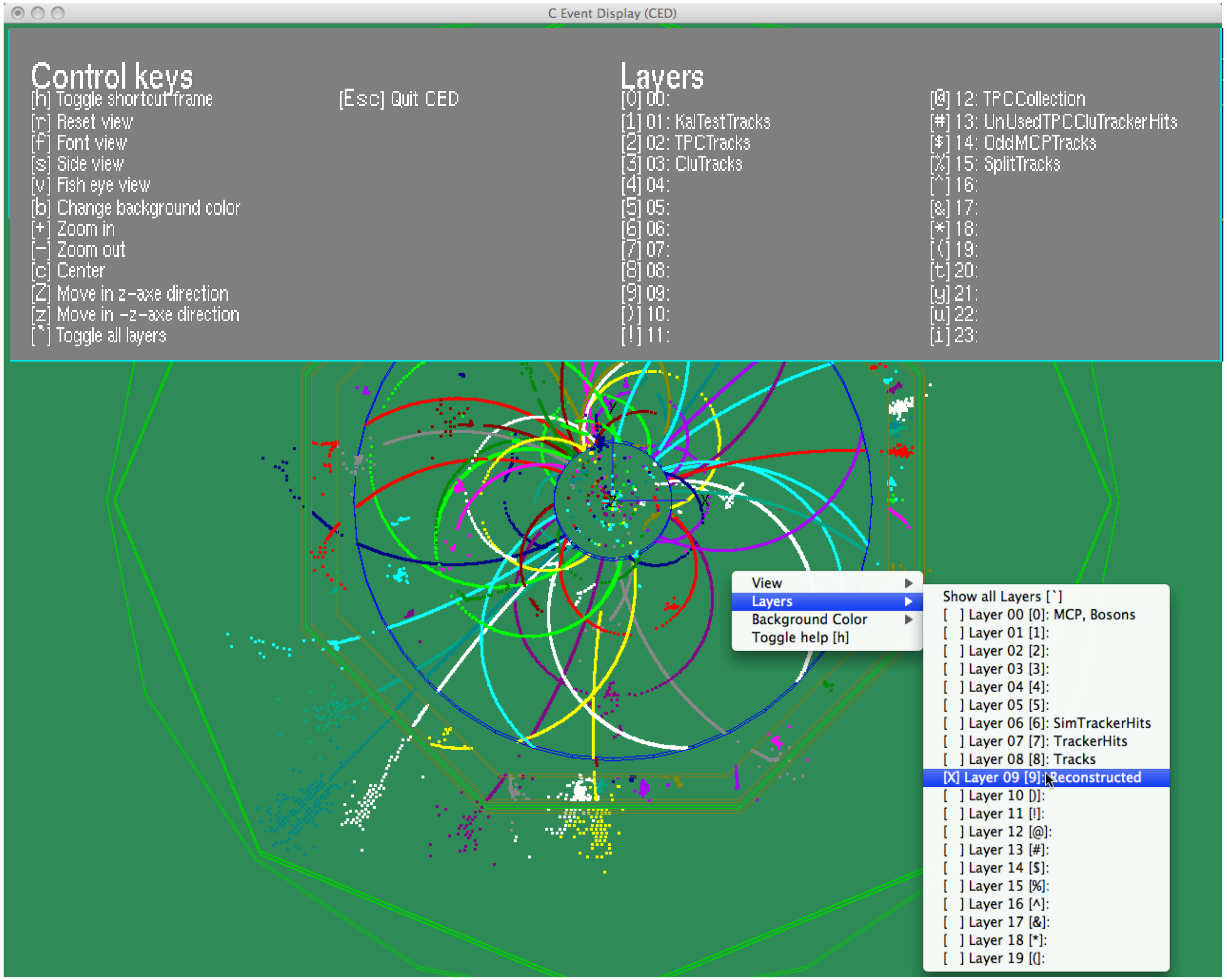
new GEAR release - v00-15



- new in v00-14(15):
  - made pure C++ (no Java)
  - improved TPCParameters for LCTPC (M.Killenberg)
    - improved global-local coordinate transformation
    - introduced circle segments in pad rows
  - Point/DistanceProperties implemented with TGeo (A.Muennich - see talk by S.Poss)
    - uses GDML interface to geant4
    - introduced material map in memory (performance)
    - **provides access to detailed geometry information !**
    - optional: Mokka-CGA (geant4) implementation of Point/DistanceProperties released in Mokka

improved CED event display

- added help menu
  - toggled with 'h'
  - shows all keys
  - shows all 'collections'
- added mouse menu
  - toggle single visualization layers
  - choose bg colors
  - views
  - zoom
- commands to add layer description and picking also for user code



```
Control keys
[h] Toggle shortcut frame          [Esc] Quit CED
[r] Reset view
[f] Font view
[s] Side view
[v] Fish eye view
[b] Change background color
[+] Zoom in
[-] Zoom out
[c] Center
[Z] Move in z-axe direction
[z] Move in -z-axe direction
[`] Toggle all layers

Layers
[0] 00:                       [@] 12: TPCCollection
[1] 01: KalTestTracks         [#] 13: UnUsedTPCCluTrackerHits
[2] 02: TPCTracks             [$] 14: OddMCPTracks
[3] 03: CluTracks             [%] 15: SplitTracks
[4] 04:                       [^] 16:
[5] 05:                       [&] 17:
[6] 06:                       [*] 18:
[7] 07:                       [(] 19:
[8] 08:                       [t] 20:
[9] 09:                       [y] 21:
[)] 10:                       [u] 22:
[!] 11:                       [i] 23:
```

- new python script to start CED & CEDViewer in on go: ced2go.py

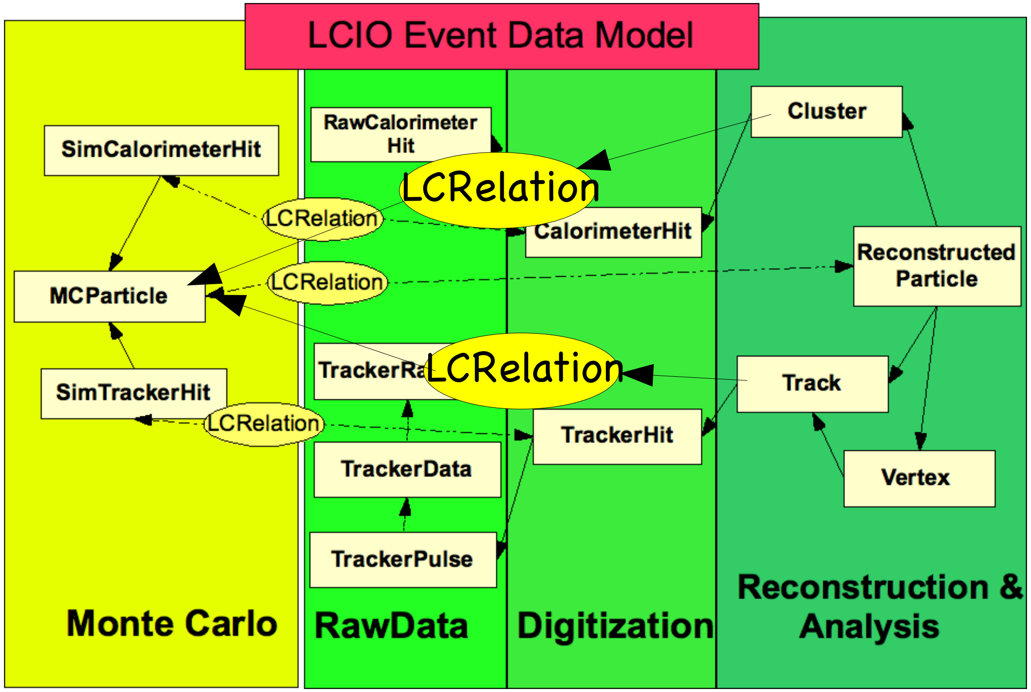
(H.Hoelbe)

Frank Gaede, IWLC 2010, CERN, Oct 18-22, 2010

MarlinReco – v00-(18)19

- new packages:
  - KinkFinder (M.Thomson, J.Marshall)
  - BCalTagEfficiency (J.List, M.Berggren)
  - FPCCDDigi (D. Kamai)
- improved packages:
  - MarlinKinfit (M.Beckmann)
  - TPCDigitizer (M.Thomson)
  - VXDDigitizer (S.Aplin)
  - SimpleMuonDigi(M.Thomson)
  - V0Finder (M.Thomson)
  - BCalTagEfficiency (C. Bartels)

- RecoMCTruthLinker
- added additional relations between MCParticle and Tracks and Clusters – to be used for DST (M.Berggren)



Frank Gaede, IWLC 2010, CERN, Oct 18-22, 2010

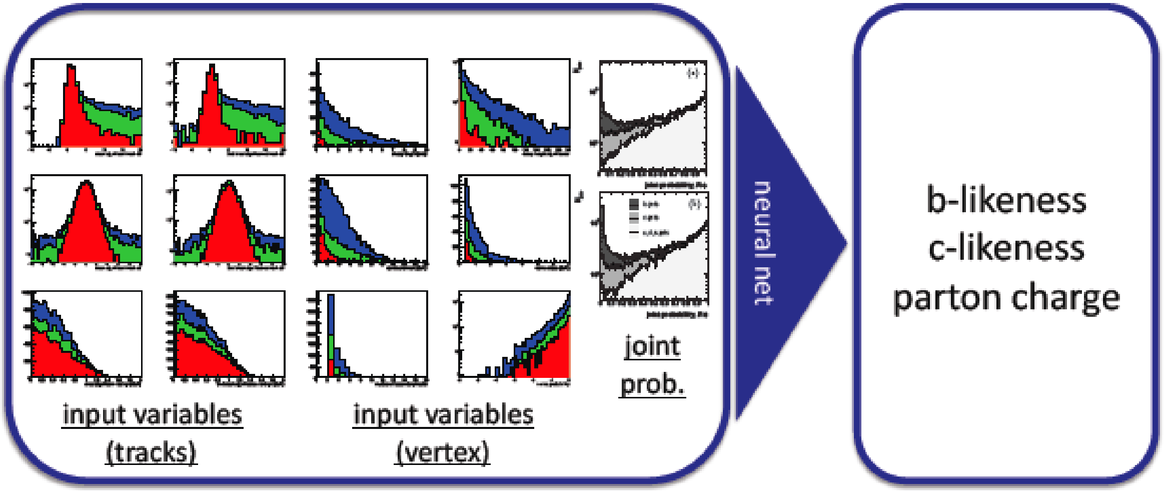
LCFIVertex

T.Tanabe, T.Suehara

from basic checks to algorithm tuning

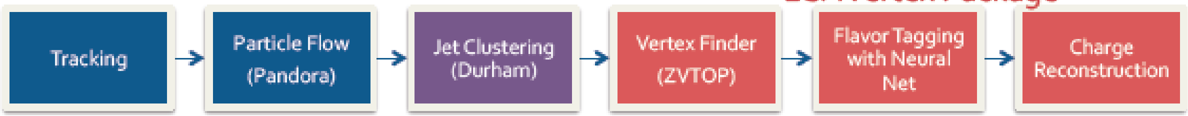
- ✓ vertex hits
- ✓ track reconstruction
- ✓ track residual & track errors
- → track quality & track selection
- → vertex reconstruction
- → combination of input variables
- → optimization for multi-jet environment

LCFIVertex is a collection of algorithms for flavor tagging and parton charge identification.

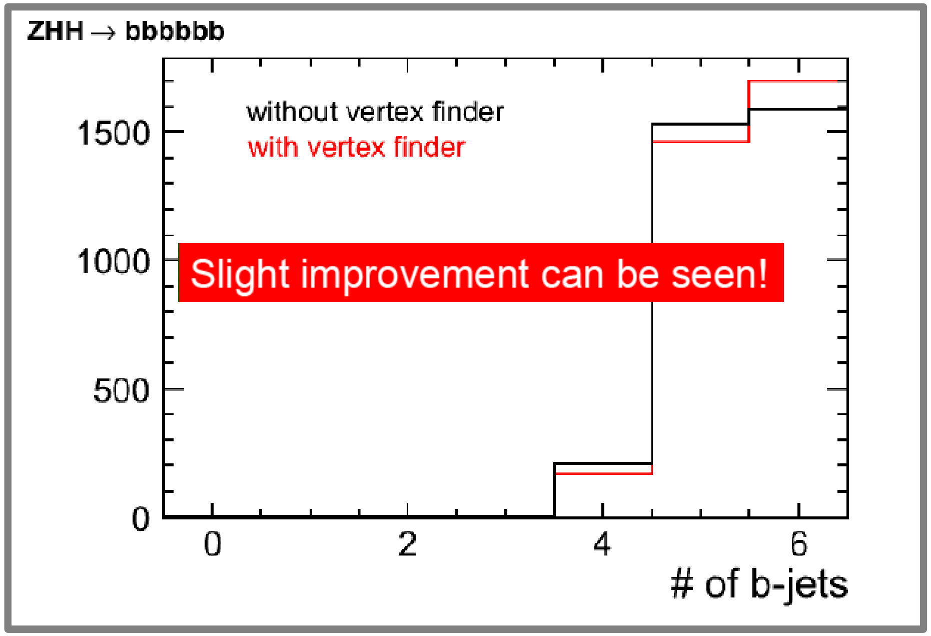


LCFI collaboration: NIM A 610 (573) [arxiv:0908.3019]

LCFIVertex Package


Tracking → Particle Flow (Pandora) → Jet Clustering (Durham) → Vertex Finder (ZVTOP) → Flavor Tagging with Neural Net → Charge Reconstruction

- japanese group taken over LCFIVertex package
- started to improve flavor tag:
    - new (kineamtic) input variables for NN
    - improve the vertexing algorithm



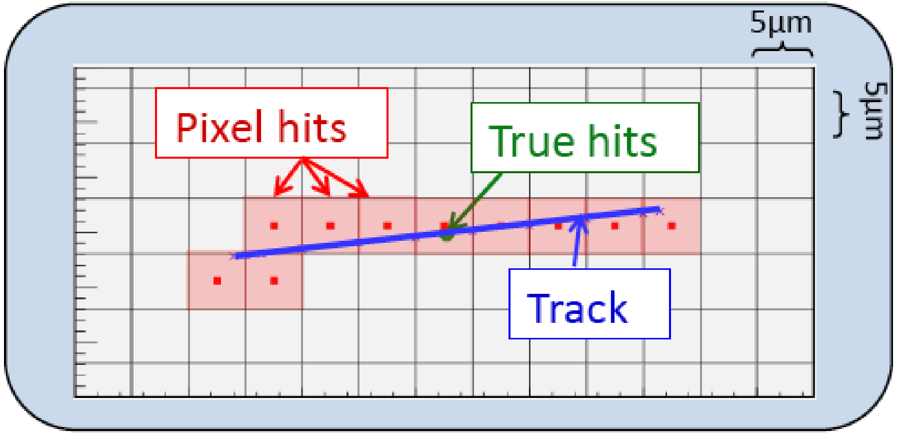
Slight improvement can be seen!

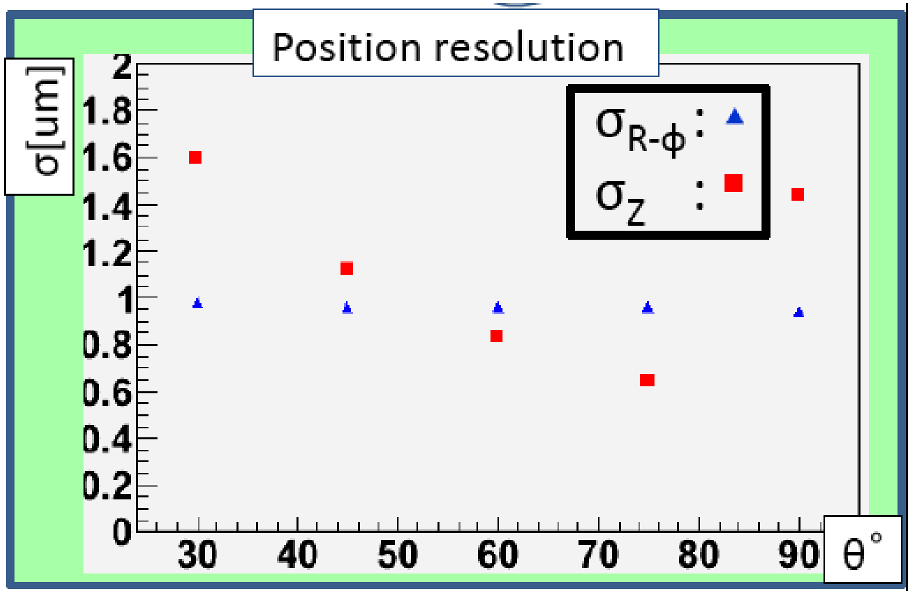
FPCCDDigitizer

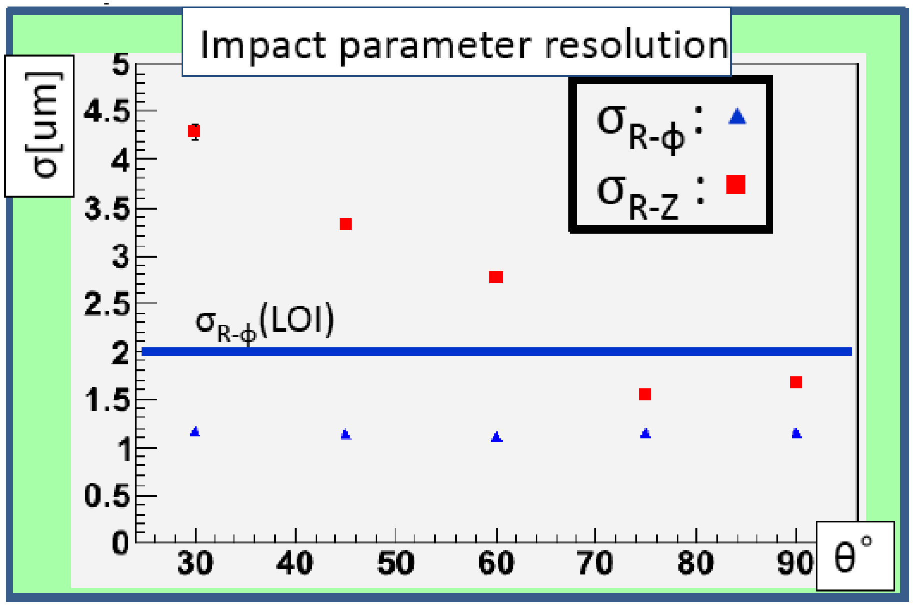
D.Kamai

FPCCD vertex detector

- FPCCD(Fine Pixel CCD)
  - Pixel size : 5µm x 5µm
  - Sensitive thickness : 15µm
- The number of pixels : ~10^{10} pixels



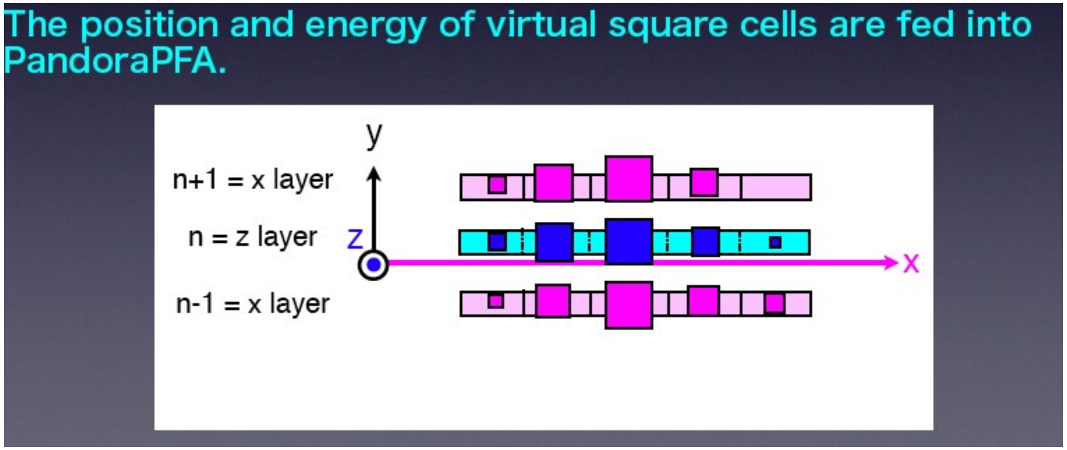




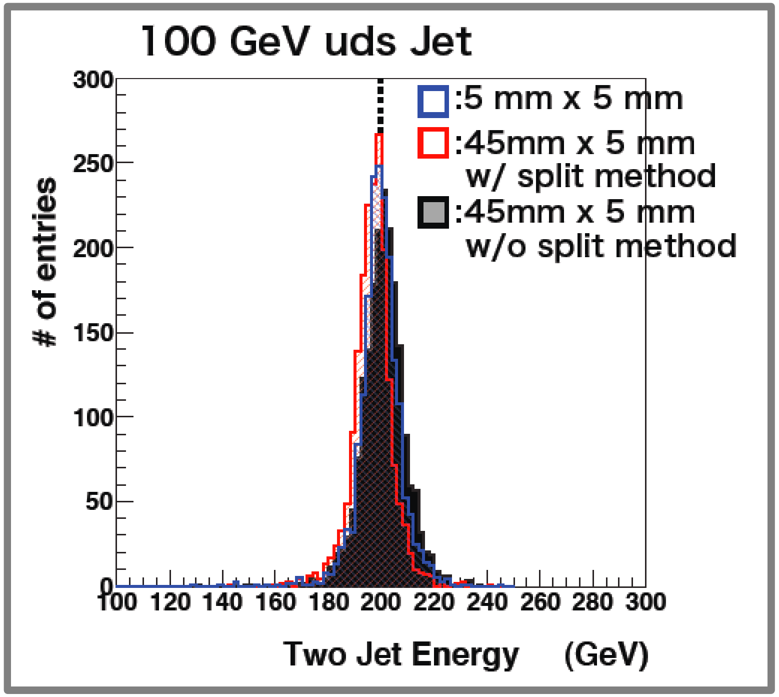
new digitizer packages released in MarlinReco for FPCCD technology option

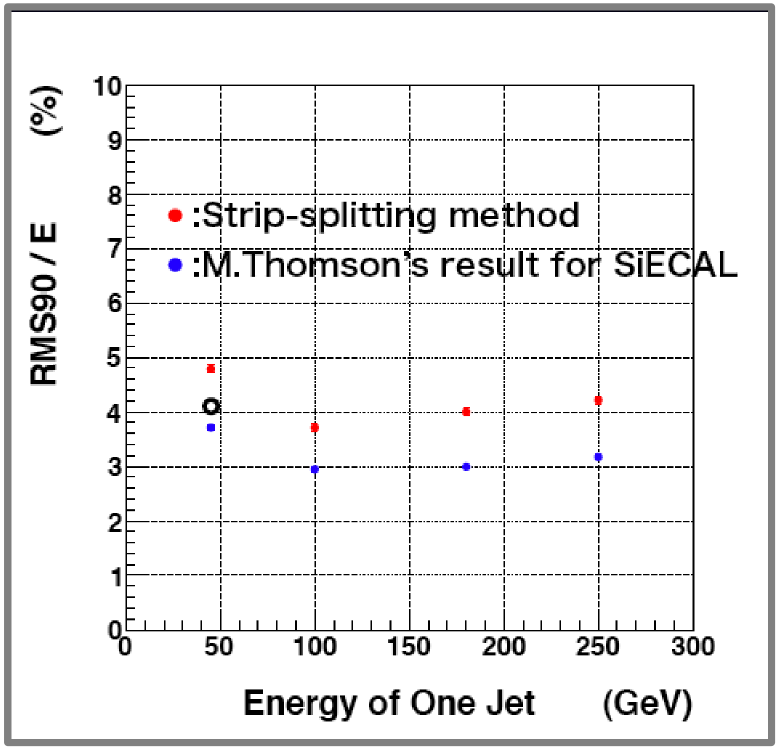
SciEcal – strip clustering

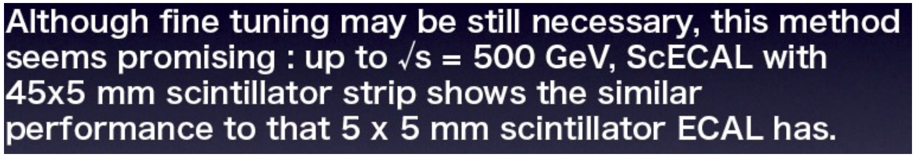
K.Coterra



The position and energy of virtual square cells are fed into PandoraPFA.





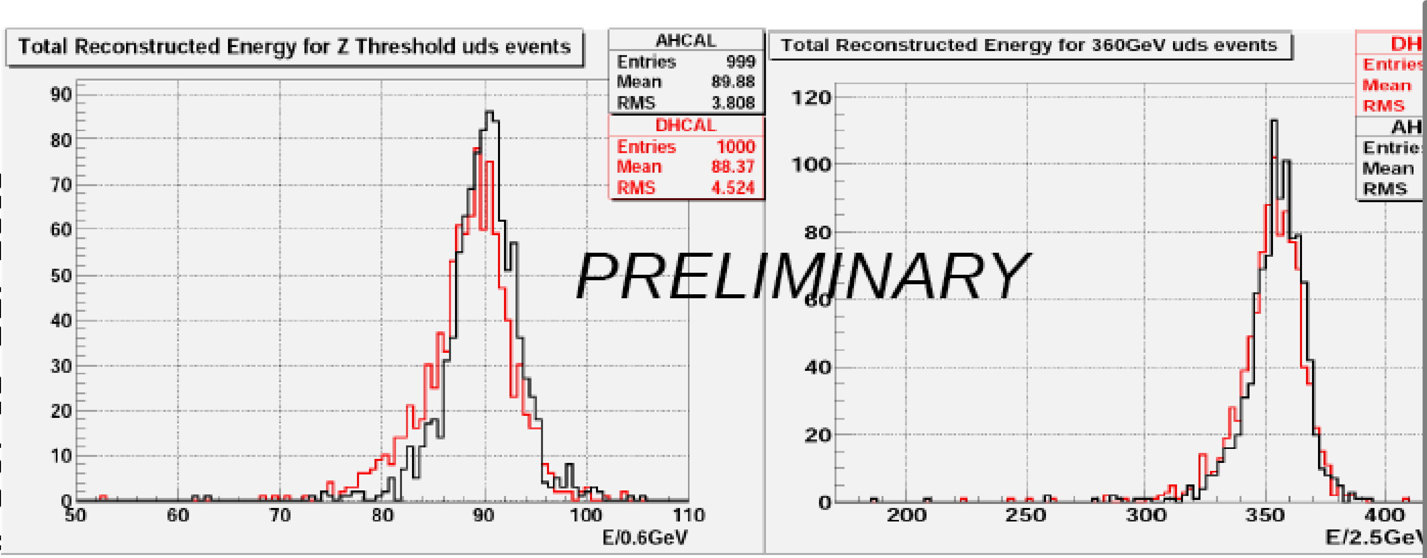


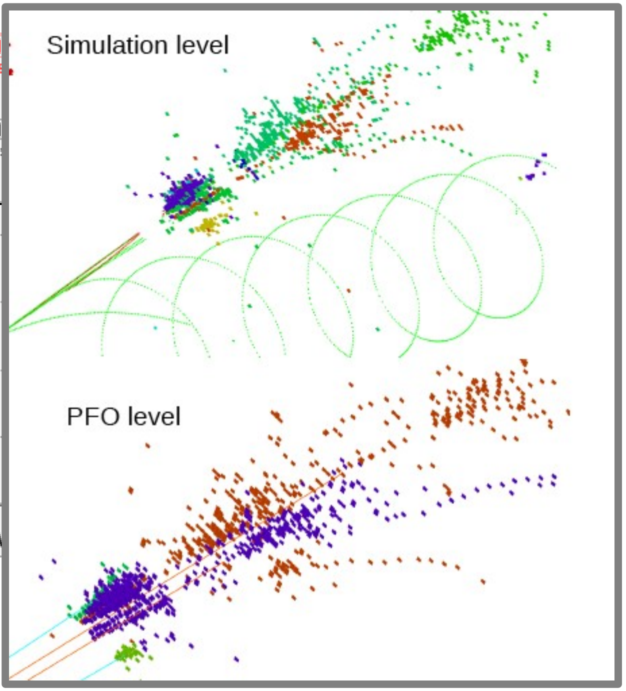
Although fine tuning may be still necessary, this method seems promising : up to √s = 500 GeV, ScECAL with 45x5 mm scintillator strip shows the similar performance to that 5 x 5 mm scintillator ECAL has.

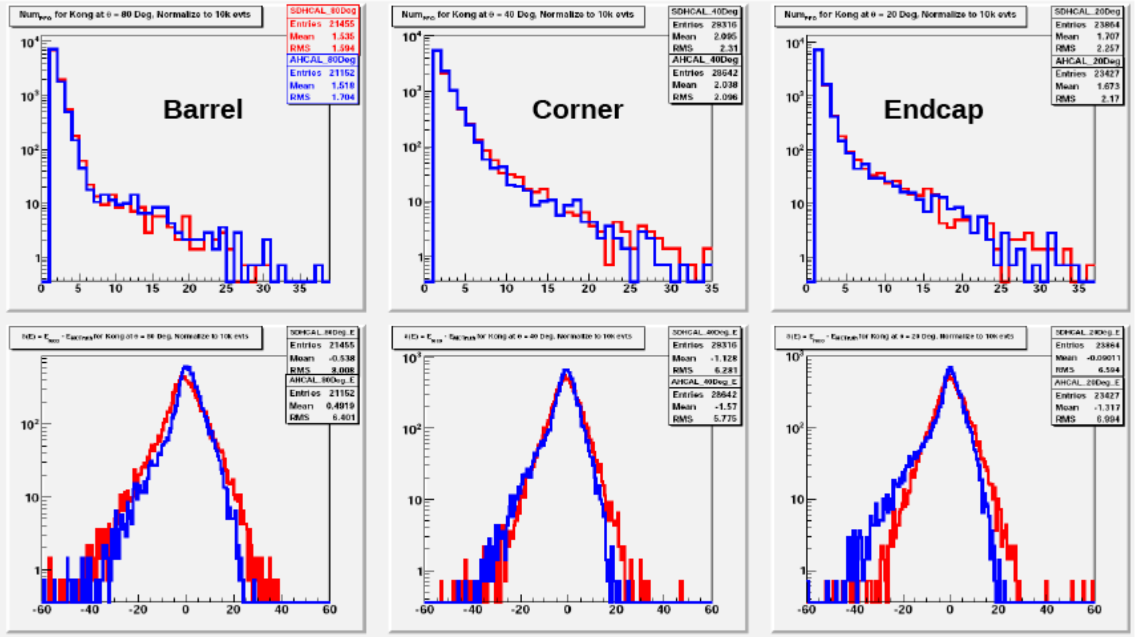
- scintillator strip clustering for SciEcal almost reaches performance of SiW Ecal with pandora

SDHcal reconstruction

M.Ruan



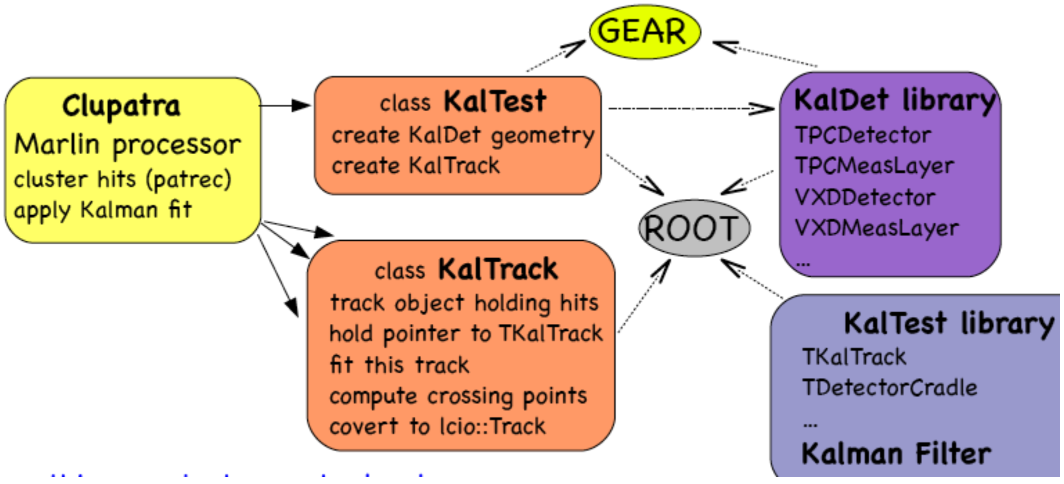




- investigating PFA performance for SDHcal compared to AHcal
- looked into details of single particle reconstruction
- pions, K0_L
- work in progress ...

Frank Gaede, IWLC 2010, CERN. Oct 18-22. 2010

new TPC tracking with KalTest



GEAR

Clupatra
Marlin processor
cluster hits (patrec)
apply Kalman fit

class **KalTest**
create KalDet geometry
create KalTrack

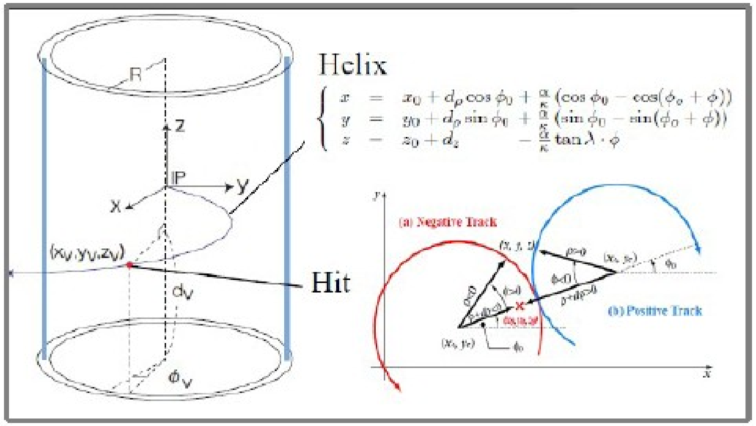
class **KalTrack**
track object holding hits
hold pointer to TKalTrack
fit this track
compute crossing points
covert to lcio::Track

ROOT

KalDet library
TPCDetector
TPCMeasLayer
VXDDetector
VXDMeasLayer
...

KalTest library
TKalTrack
TDetectorCradle
...
Kalman Filter

- included KalTest Kalman filter (K.Fujii) in iLCSoft - to be released in v01-10
- used by LCTPC and ILD/iLCSoft
- have loose coupling to Marlin pattern recognition (based on NN-Clustering)
- use for track fit and track extrapolation



Helix

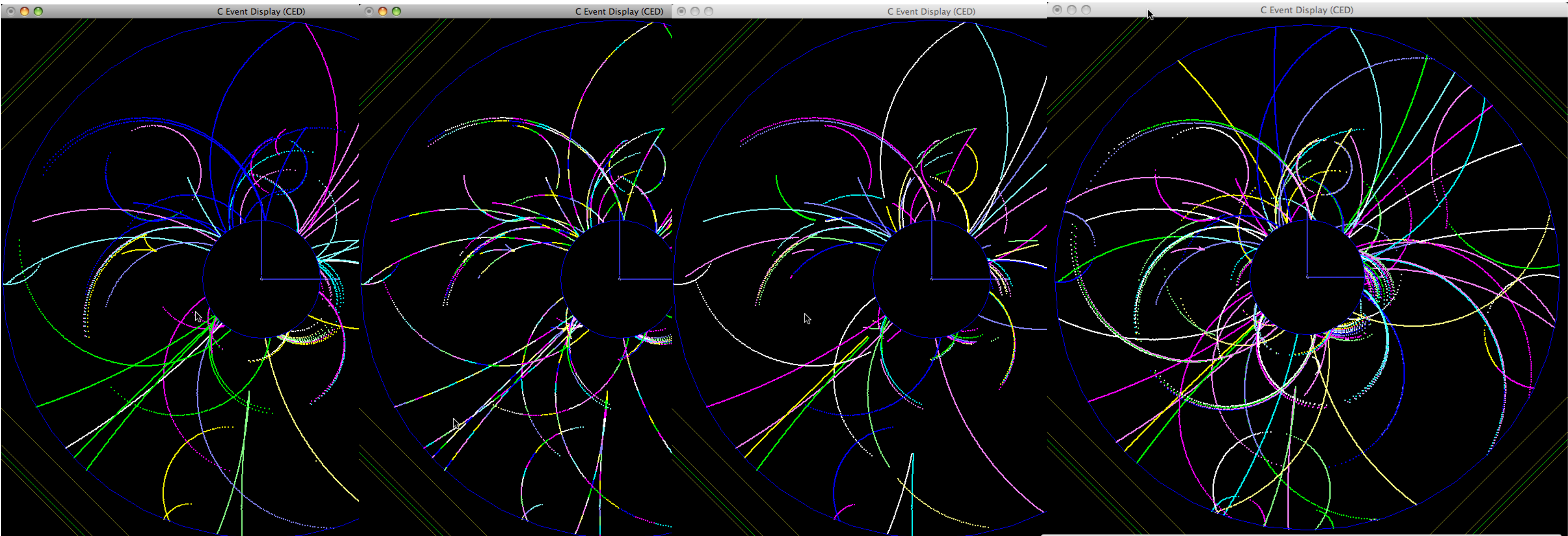
$$\begin{cases} x = x_0 + d_\rho \cos\phi_0 + \frac{\alpha}{\kappa}(\cos\phi_0 - \cos(\phi_0 + \phi)) \\ y = y_0 + d_\rho \sin\phi_0 + \frac{\alpha}{\kappa}(\sin\phi_0 - \sin(\phi_0 + \phi)) \\ z = z_0 + d_z - \frac{\alpha}{\kappa}\tan\lambda \cdot \phi \end{cases}$$

- track parameters correspond to LCIO, except:
  - d0_lcio = - drho_kaltest
  - omega_lcio = a(cB) * kappa_kaltest
- and different units:
  - KalTest: cm, KGauss, GeV
  - LCIO: mm, Tesla, GeV

plan to adapt Kaltest units to LCIO

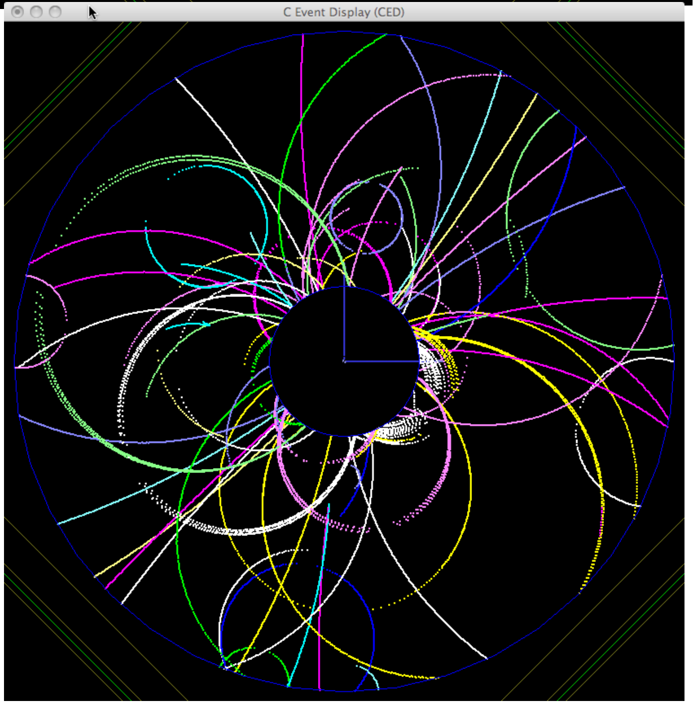
6

TPC patrec with NN-clustering



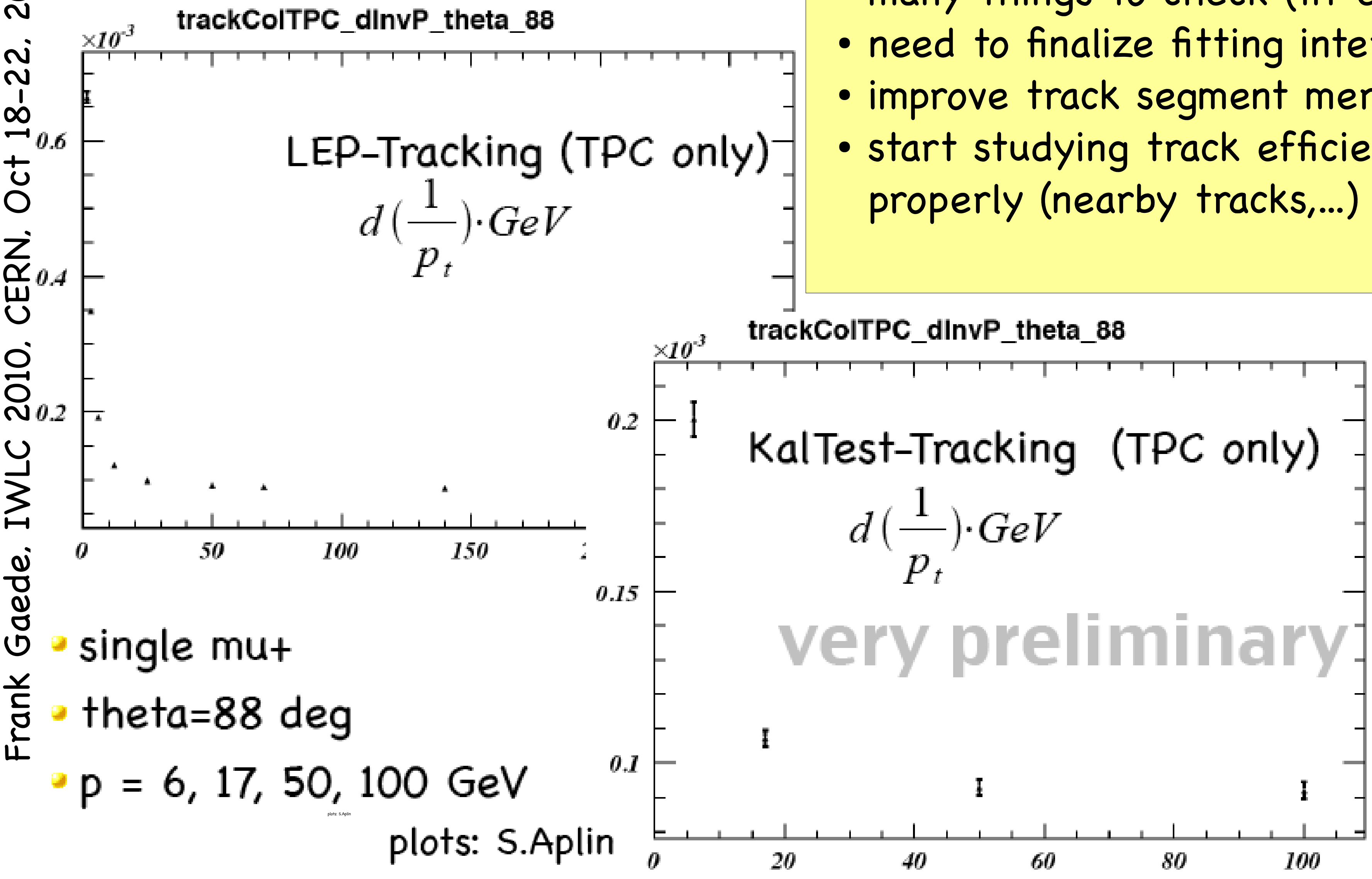
- cluster hits with nearest neighbor condition
  - identify 'merged' tracks
  - re-cluster in small pad row ranges
  - remove pad row ranges with overlap
- use KalTest fit to compute crossinging points
  - assign leftover hits to best matching track
- merge resulting track segments (e.g. R, x_c, y_c)

-> need to study efficiencies systematically



first 'results' using KalTest fit

- momentum resolution for single muons looks quite promising
- many things to check (fit errors)...
- need to finalize fitting interface
- improve track segment merging
- start studying track efficiencies properly (nearby tracks,...)



- single mu+
- theta=88 deg
- p = 6, 17, 50, 100 GeV

plots: S.Aplin

Summary – core & reco software

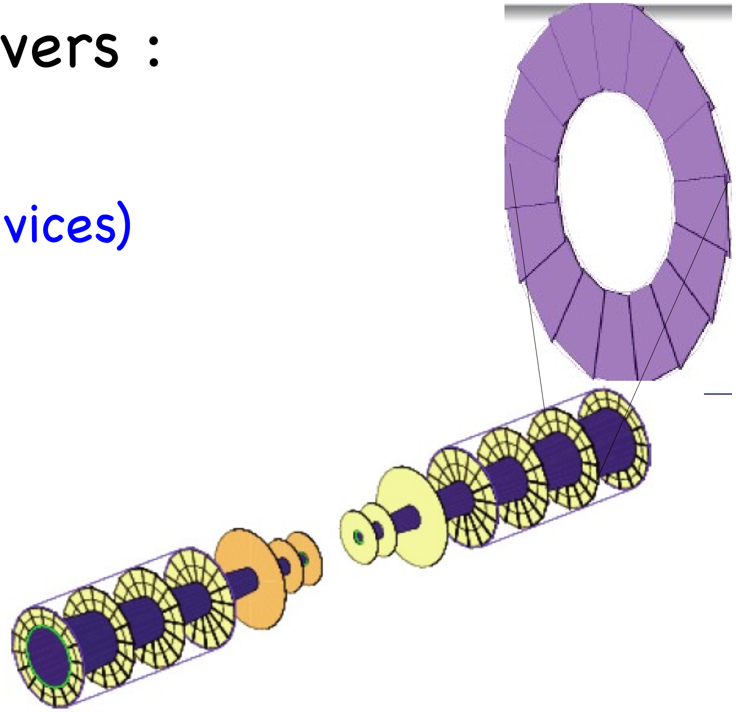
- new developments in core software:
  - LCIO v01-51: direct access, ROOT dictionary, improved EDM
  - GEAR: access to detailed geometry and material through TGeo interface
  - CED: added picking, helper menu, mouse controlled menu
- reconstruction software
  - MarlinReco: many (small) improvements by many people
  - Clupatra: started development of TPC patrec using KalTest fitter
  - LCFIVertex: started to improve the flavor tag
  - FPCCD: new digitizer package
  - SciEcal: strip clustering improved
  - SDHcal: started to investigate PFA performance

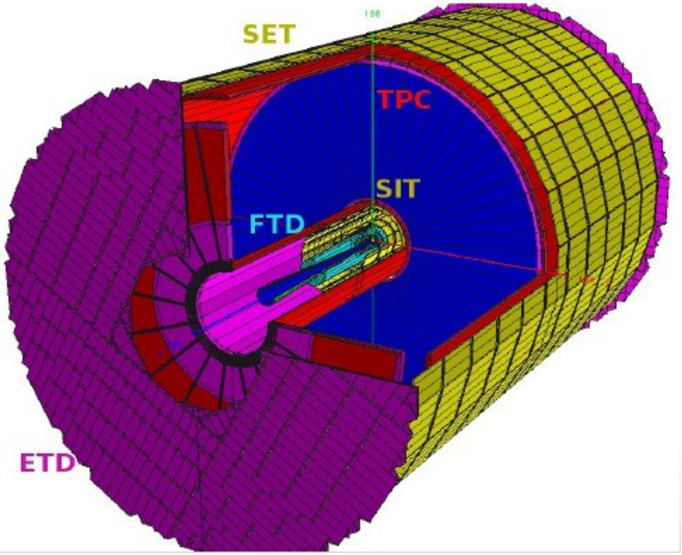
=> considerable new developments in reconstruction code in particular great to see that now technology options are addressesed in the software

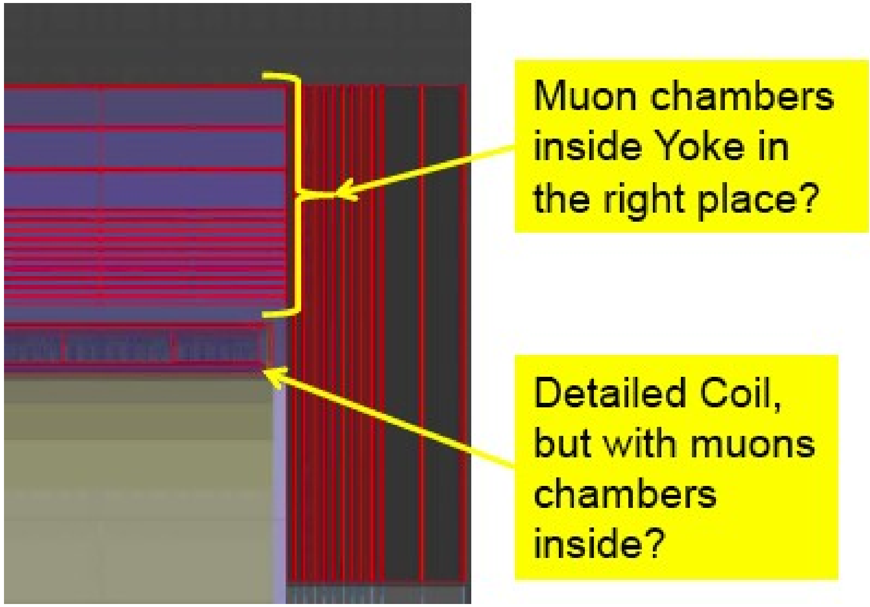
new Mokka release – towards ILD_01

P.Mora de Freitas

- major rewrite of some sub detector drivers :
  - SIT, SET, ETD - FTD - Muon
  - increased level of detail and realism (incl. services)
  - driver for overall services for TPC, Ecal, Hcal
  - added electronics boards to ECal and Hcal







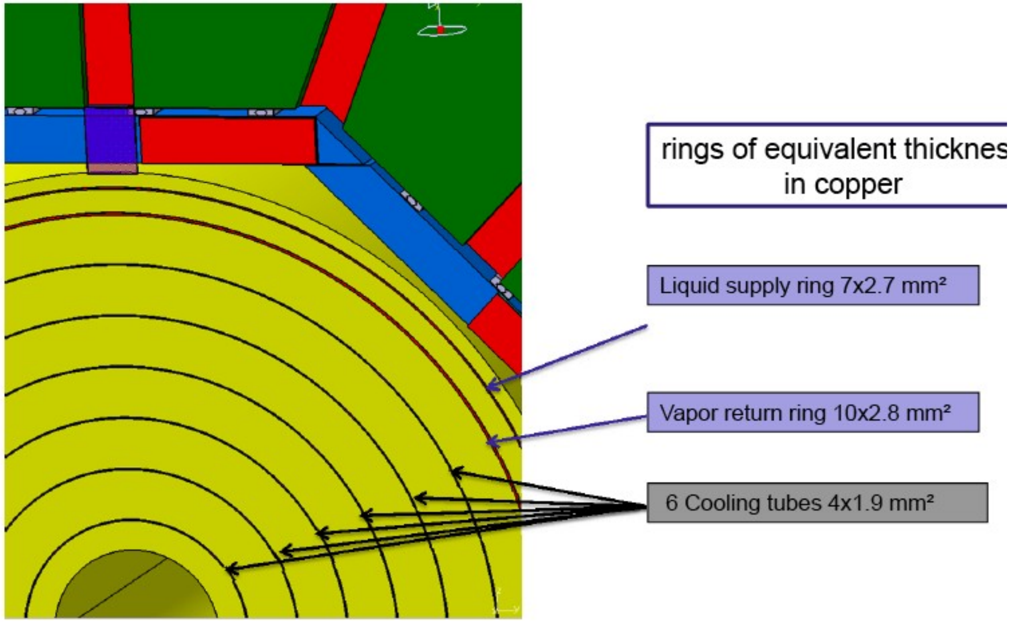
Muon chambers inside Yoke in the right place?

Detailed Coil, but with muons chambers inside?

work of many people:
A.Charpy, J.Duarte, A.Saveliev, G.Musat, A.Lucaci, P.Mora de Freitas,....

new Mokka release - towards ILD_01

- added cabling and services for TPC, ECal & Hcal (C.Clerc, G.Musat)
- still missing: inner detector services (to be defined by R&D groups)

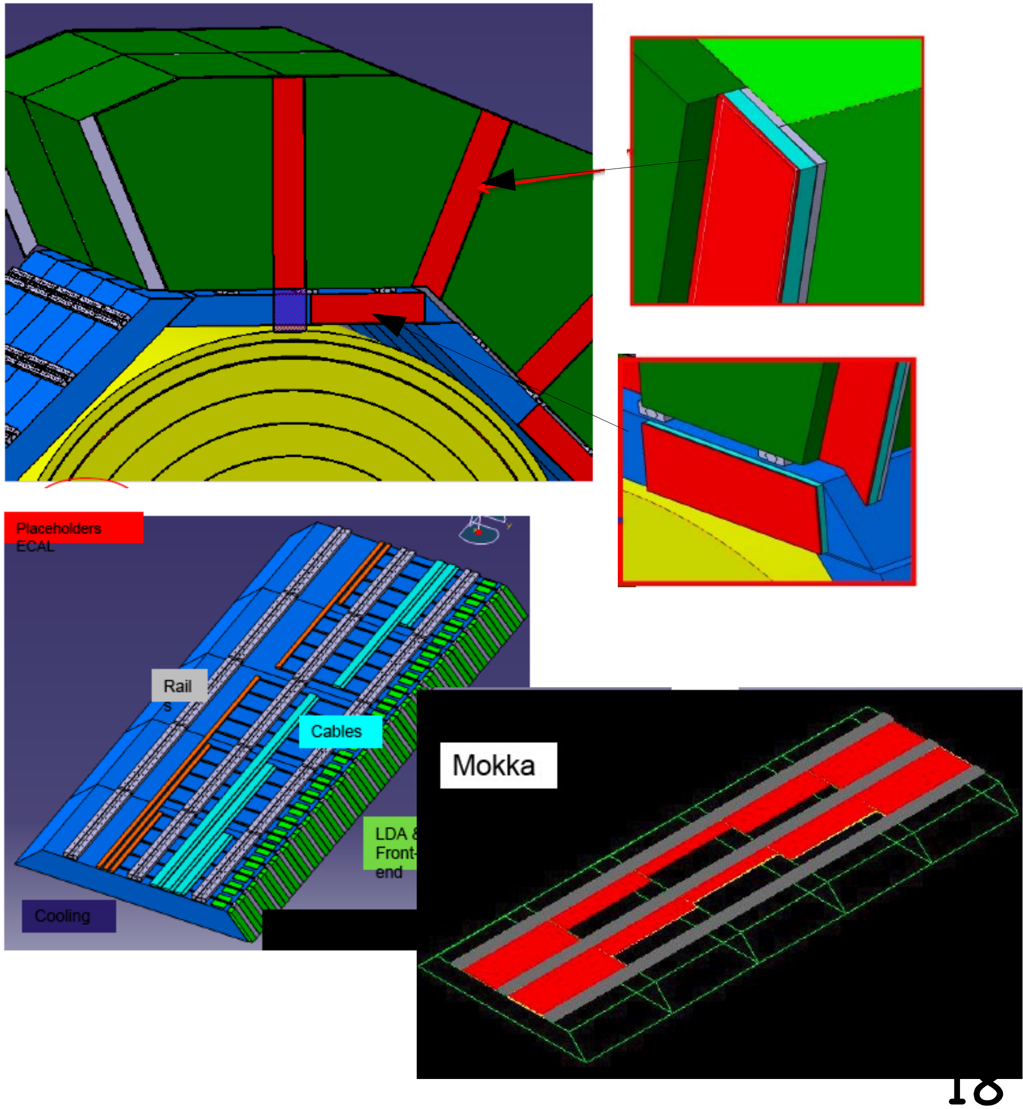


rings of equivalent thicknes in copper

Liquid supply ring 7x2.7 mm²

Vapor return ring 10x2.8 mm²

6 Cooling tubes 4x1.9 mm²



big step forward in increasing realism of ILD detector simulation !

Frank Gaede, IWLC 2010, CERN, Oct 18-22, 2010

ILD simulation status overview

- beam pipe:
  - no final engineering design
  - first design exists
- B-field
  - realistic field map for bg studies
  - simple field for mass production
- Physics List
  - use QGSP_BERT (re. by geant4)
  - issues in tungsten @CLIC !?
  - possible other - FTF (results from Calice)
- VXD
  - realistic models for 3 double and five single layers
  - cabling missing
  - first estimate of services exist
- FTD
  - more realistic design implemented in Mokka
- SIT, SET, ETD
  - realistic and detailed sim. exists
  - now implemented in Mokka
- TPC
  - rather realistic simulation
  - cabling and support implemented

red: done since Simulation Workshop 2010 DESY
blue: ongoing work or to be addressed

ILD simulation status overview

- Sci- and Si/W ECal
  - realistic driver exists
  - can vary mix of Scint./Silicon
  - -> can study options !
  - cabling and services implemented
- dHcal - Sci Hcal
  - realistic simulation drivers exists
  - two geometries for dHCAL
  - cabling and services implemented
- Muon
  - new more realistic model exists
  - needs to be verified
  - strips vs. tiles ?
  - instrumented coil ?
- BeamCal
  - new engineering design exists
  - implemented in Mokka
- LCal
  - new realstic driver exists
  - including support, cooling
- LHCAl
  - no real design exists

towards a simulation baseline ILD_01

- a new Mokka model ILD_01 to be released soon
  - including the new drivers for SIT, SET, ETD, FTD, Muon, Services
  - other drivers as in ILD_00 (the LOI model)
  - will have the required realism for the DBD !?
- need to debug, test and check this model

- need to develop new reconstruction code:
  - digitization for new Si-tracking detectors
    - strips, ghost hits, smear on wavers (as opposed to cylinders and disks)
  - develop entirely new tracking code for these detectors
- => this is a major effort and will take some time

=> need to use the well understood ILD_00 model for detector optimization and for studying technology options

towards the simulation baseline

- the ILD_01 model will be a first iteration towards the simulation baseline detector model to be used for the DBD
  - it is an evolution of the LOI model ILD_00

- however it is of course not a decision on which technology options should be used for the DBD
- this will have to be done by ILD !
- how and when should we do this ?

- could start now
- need to do this at LCWS2011 - the latest
- or have dedicated ILD meeting in summer 2011

  like Cambridge 2008

aditional material

a ROOT dictionary for LCIO

- LCIO now comes with a ROOT dictionary for all LCIO classes (optional) - with this one can:

  (allready in v01-12-01)

  - use LCIO classes in ROOT macros
  - write simple ROOT trees, e.g. std::vector<MCParticleImpl*>
  - use TTreeDraw for quick interactive analysis of LCObjects:

```
//---gamma conversions:
TCut isPhoton("MCParticlesSkimmed.getPDG()==22" ) ;
LCIO->Draw("MCParticlesSkimmed._endpoint[][0]:
            MCParticlesSkimmed._endpoint[][1]",isPhoton ) ;
```

  - write complete LCIO events in one ROOT branch
  - see: $LCIO/examples/cpp/rootDict/README for details & help

- -> we are interested in feedback from the users if this provides already the requested features

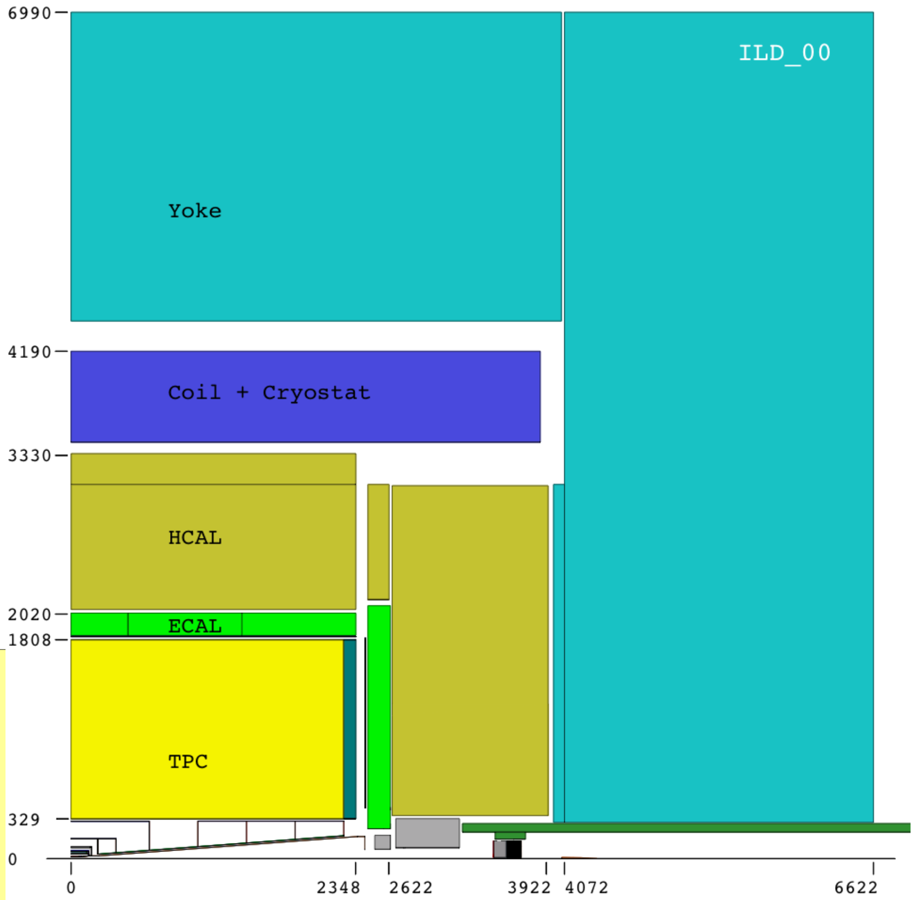
Digitization & Reconstruction in Marlin

- VXD, SIT, FTD, SET, ETD
  - smearing of 3D space points according to detector resolutions as established by R&D groups
- TPC hits
  - smearing of 3D space points - taking into account drift distance, polar and azimuthal angle of track
  - parameterization from TPC R&D groups
- ECal, HCal, LCal, Bcal, LHCal, Muon Calo hits
  - calibration (single particle resolution)

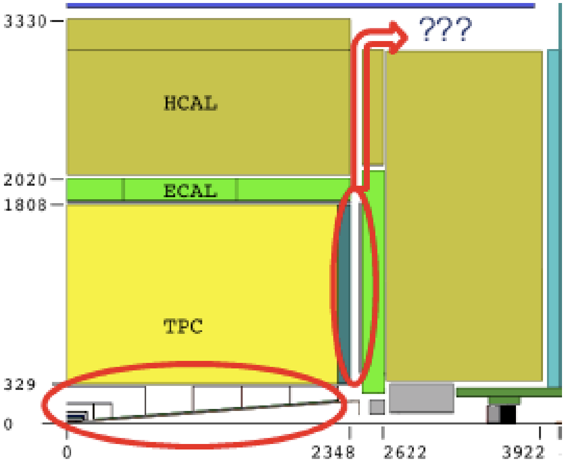
- Tracking
  - standalone tracking in Silicon detectors and TPC - **MarlinReco-FullLDCTracking**
  - Kalman filtering: wrapped f77 from LEP
  - => need for new tracking code - development started
- Particle Flow Algorithm
  - **MarlinPandora/PandoraPFANew**
- JetFinder
  - Durham jet finder (run for 2-6 jets)
- Flavour Tagging
  - **LCFIVertex** package: ZVTop, ZVRes + Neural Network Fl.Tag
- DST Maker
  - ReconstructedParticles, Jets, Tracks and Clusters (25k/evt)

Mokka Simulation ILD

- defined 'ILD simulation reference model' for LOI mass production
- engineering level of detail for most subdetectors:
  - support structures
  - cracks
- modified to ILD_CLIC detector model to be used for CLIC CDR !



- goal: further improve realism of ILD model for **DBD (ILD_01)**
  - faults and imperfections
  - cables and services
  - level of detail
  - ...


3330
???
HCAL
2020
ECAL
1808
TPC
329
0
0
2348
2622
3922