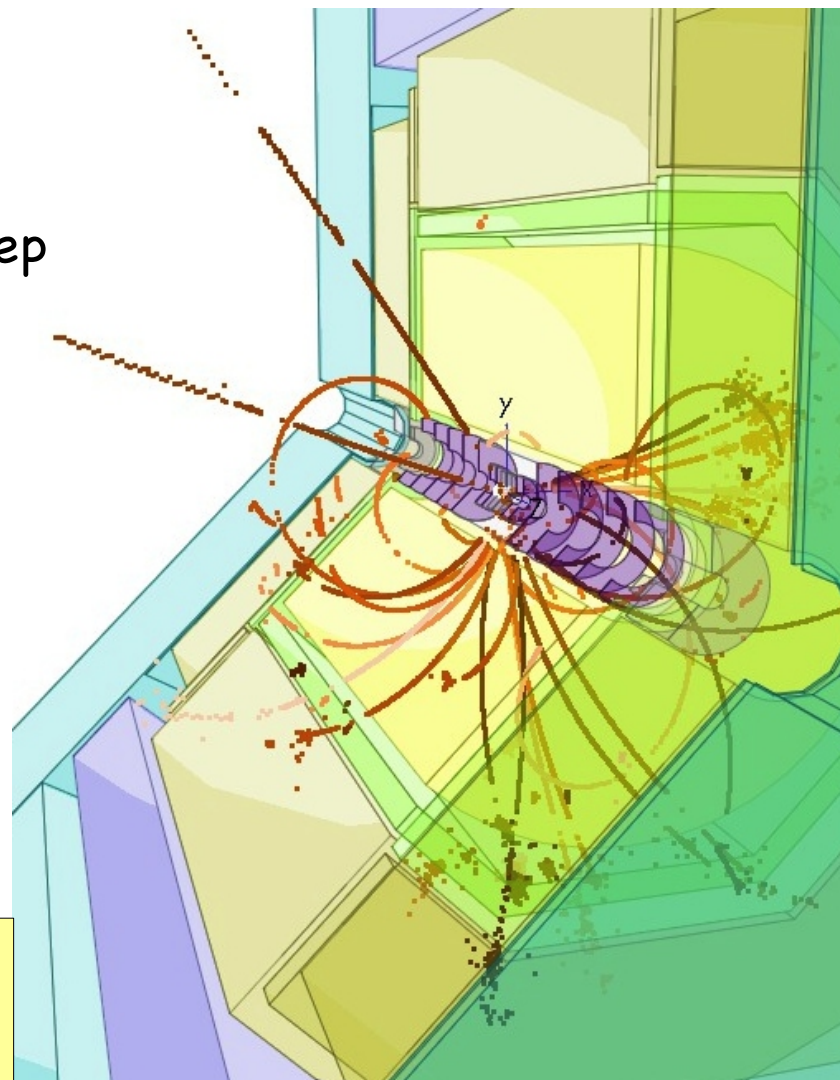


Progress in DD4hep based simulation and reconstruction

Frank Gaede, DESY/CERN
ILD Software and Analysis Meeting
Nov 5, 2014

Outline

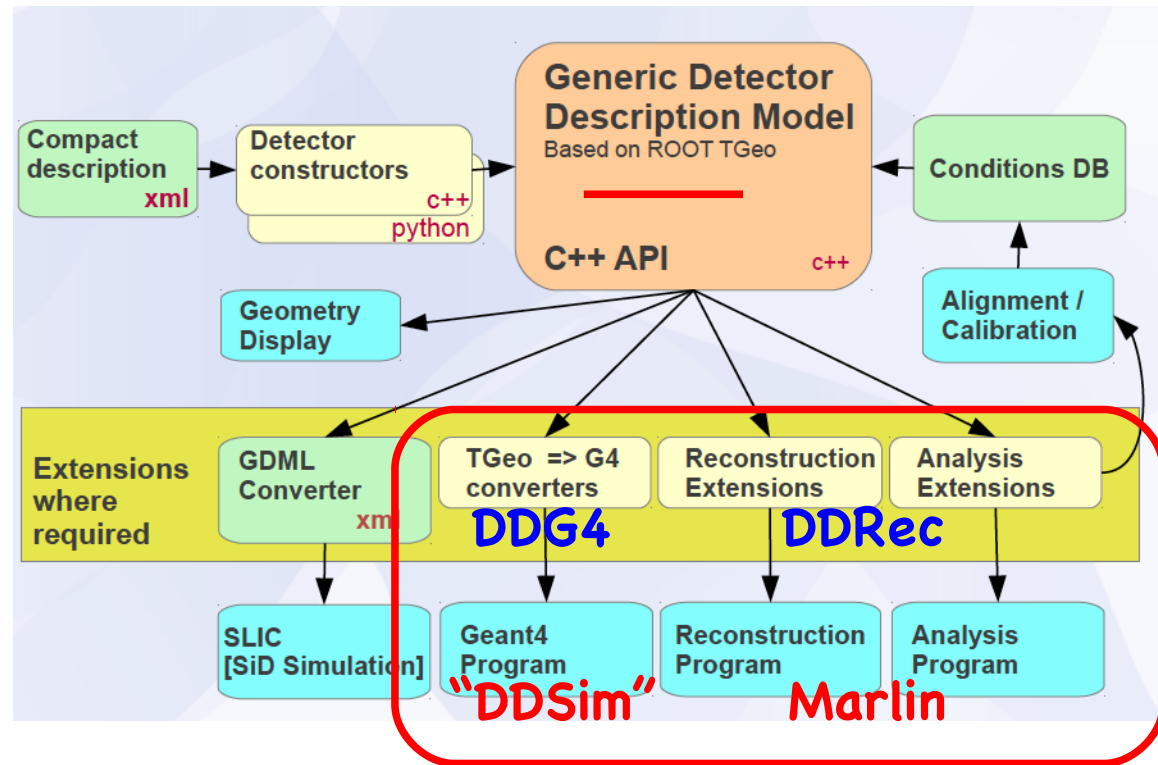
- Introduction
- DDSim
 - new ILD simulation model in DD4hep
- DDRec
 - interface to Reconstruction
 - interface to GEAR
- running MarlinReco et al
- Summary



report on work by:
[M.Frank](#), C.Grefe, A.Sailer, N.Nikiforou,
S.Lu, M.Petric, F.G.

DD4hep - overview

- **DD4hep** common detector geometry description
 - developed in AIDA WP2 (CERN, DESY)
 - used by CLICdp, FCC and ILD
- advantages of DD4hep:
 - better, more consistent description of detector geometry with one unique source
 - possibility to simulate misalignment to study **alignment** strategies for ILD
 - cooperation w/ CLICdp (and SiD)

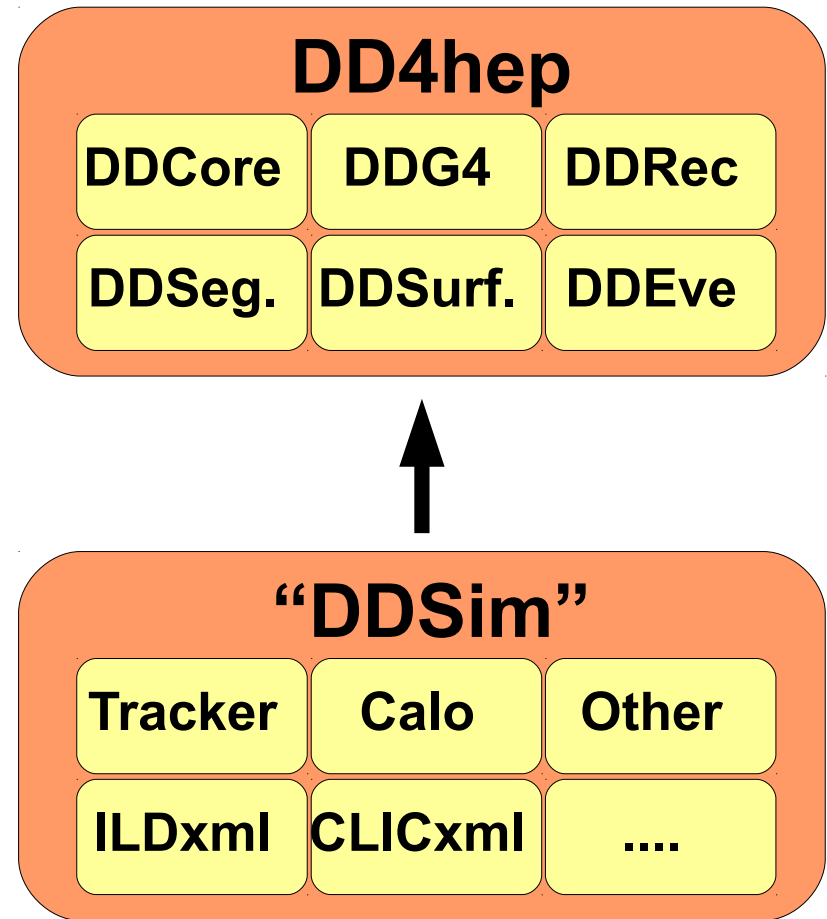


- **DD4hep** sub packages:
 - **DDG4**: geant4 gateway -> simulation
 - **DDRRec**: interface for reconstruction (Gear++)
- ilcsoft packages:
 - **DDSim** (to be renamed) -> detector description and interface to simulation
 - **Marlin**: reconstruction and analysis

the "DDSim" package

- created package "**DDSim**" as a **common LC detector description package** for ILD and CLIC (and SiD) that is used for simulation and reconstruction
- eventually want to preserve the current Mokka models - started with ILD_o1_v05:
 - extract DB params to xml and
 - line-by-line port of geometry drivers
- will soon add CLIC models from DD4hep/examples
- **eventually develop new and better ILD models**
- where possible have concept agnostic models (ILD and CLIC)

<https://svnsrv.desy.de/viewvc/ddsim/DDSim/>



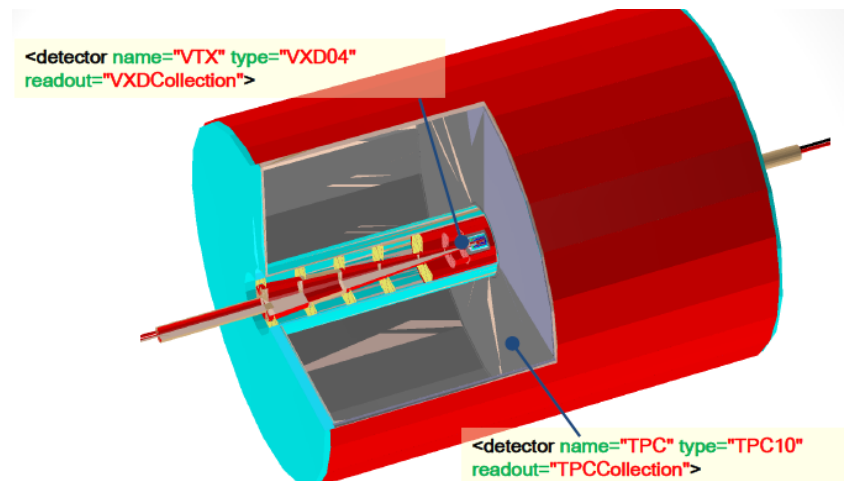
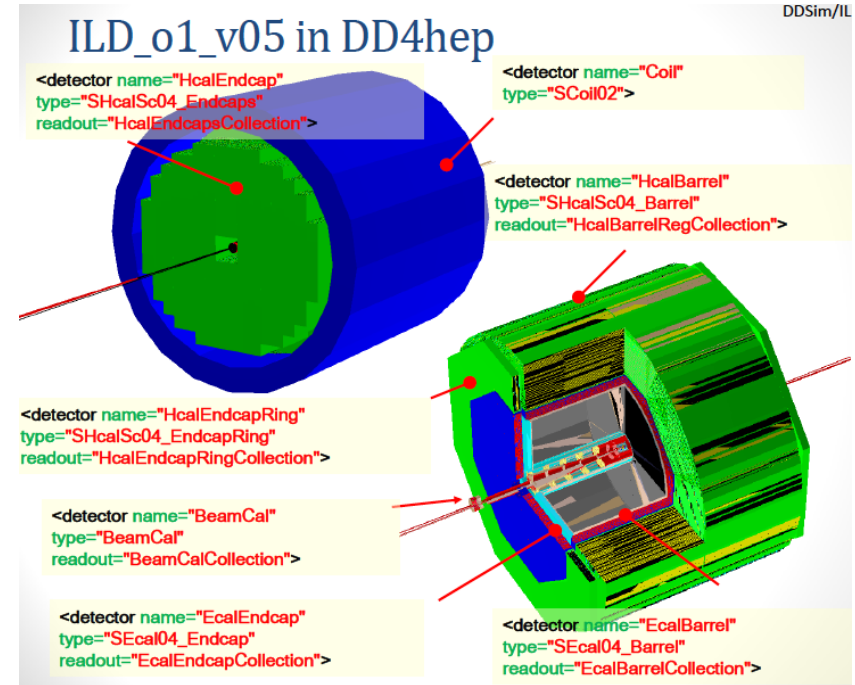
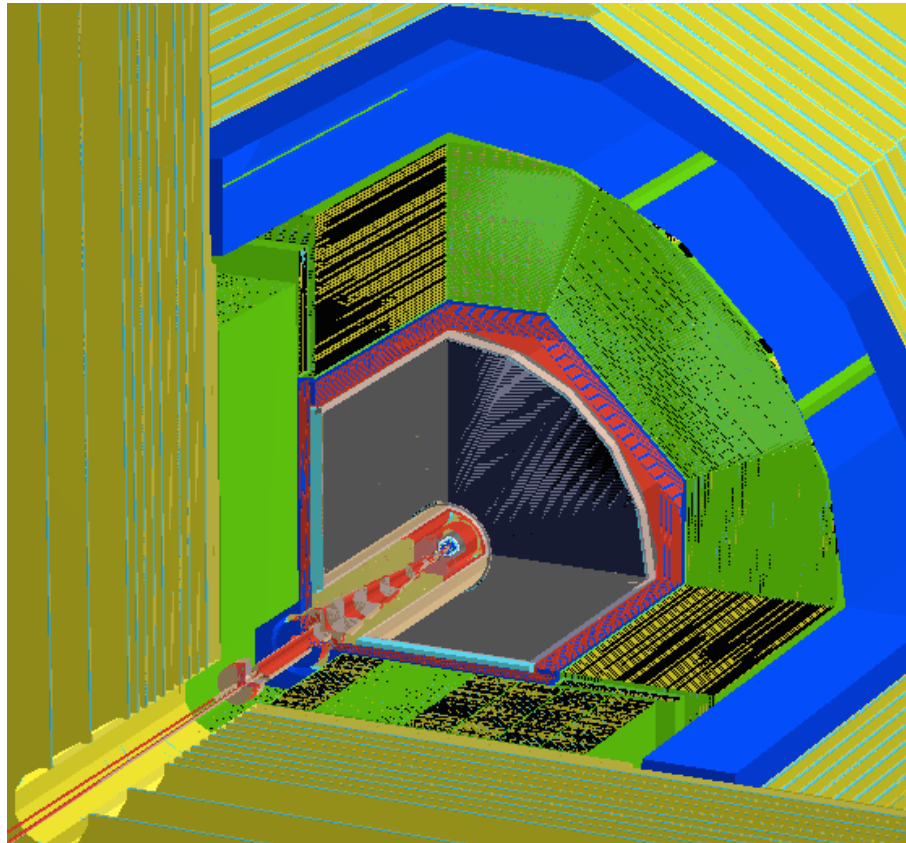
DDSim should be renamed - suggestions welcome, eg:

LINear collider **D**Etector **G**e**O**metry

Linear **C**ollider detector **G**e**O**metry 4

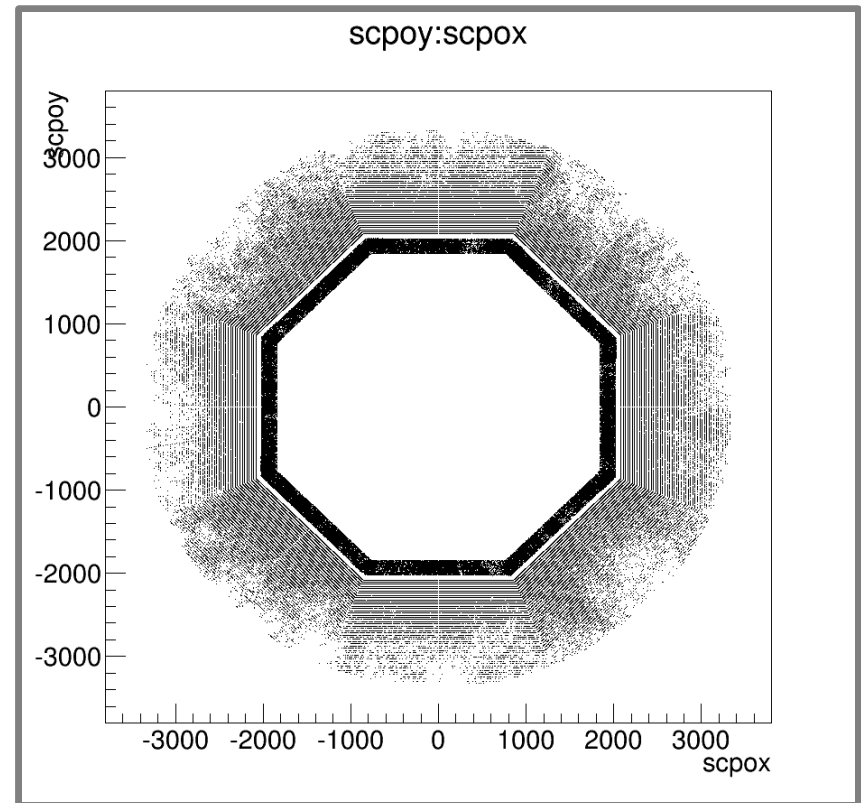
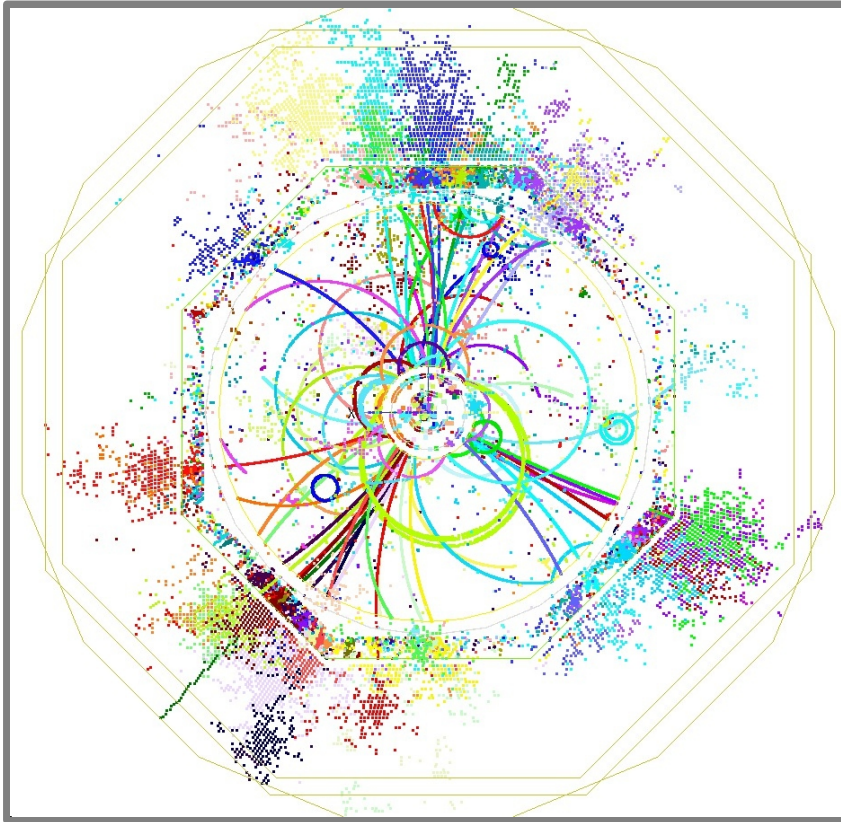
ILD_o1_v05 in DDSim

- complete Mokka model ILD_o1_v05 ported:
- VXD, FTD, SIT, TPC, SET, beam pipe
- Ecal, Hcal, Yoke
- Beamcal, Lcal, LHcal
- services (started w/ TPC cooling)



checking hits in new model I

- have two generic SensitiveDetector implementations for trackers and calorimeters that can be used for *very first* tests



- SimHits for ttbar @ 500 GeV
- hits colored with MC-truth information in Sim-Hits

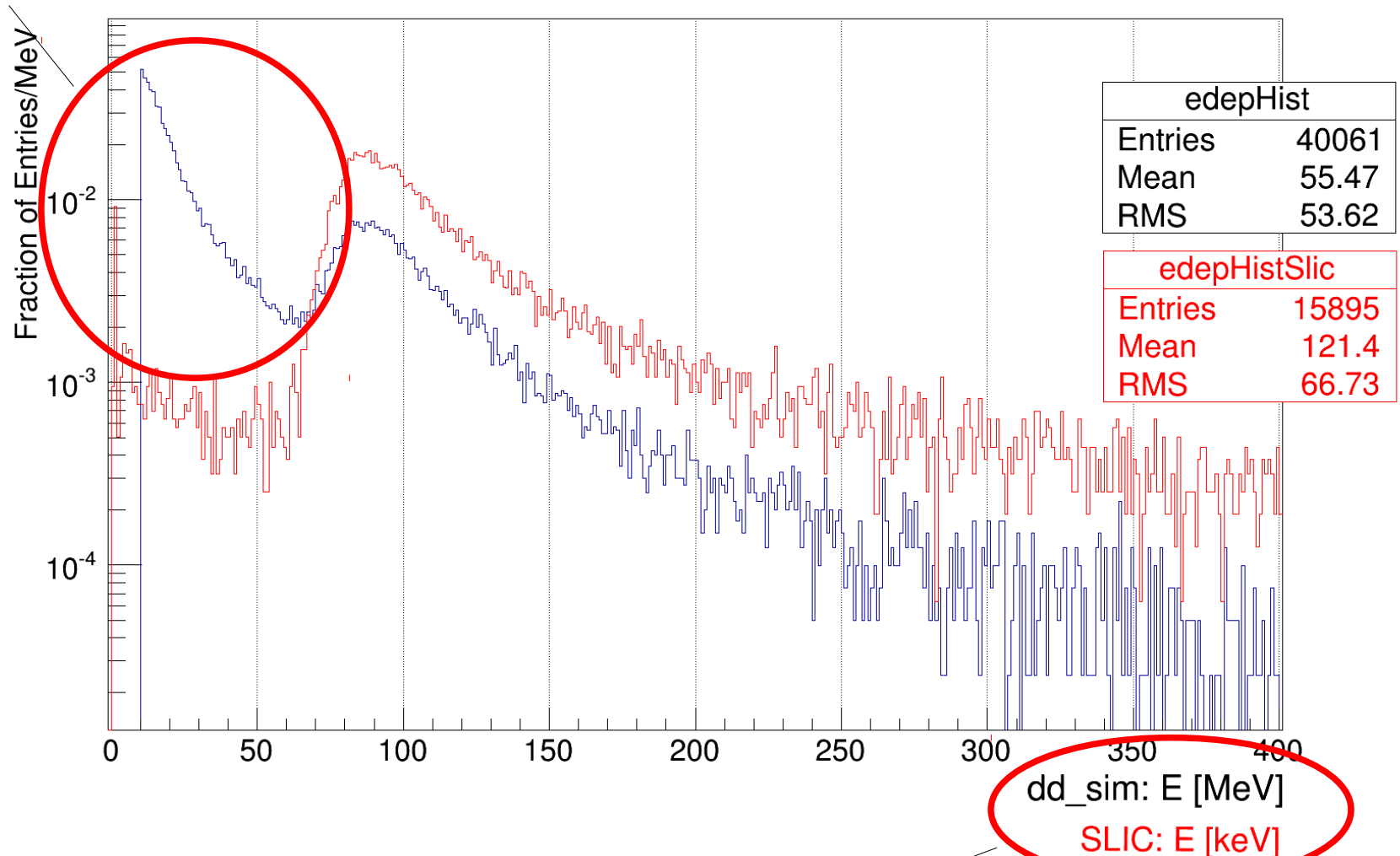
- SimHits for ttbar @ 500 GeV
- hit map for Ecal and Hcal barrel calorimeters

checking hits in new model II

possible artifact
of the current
sensitive detector
creating too
many hits ?

N.Nikiforou

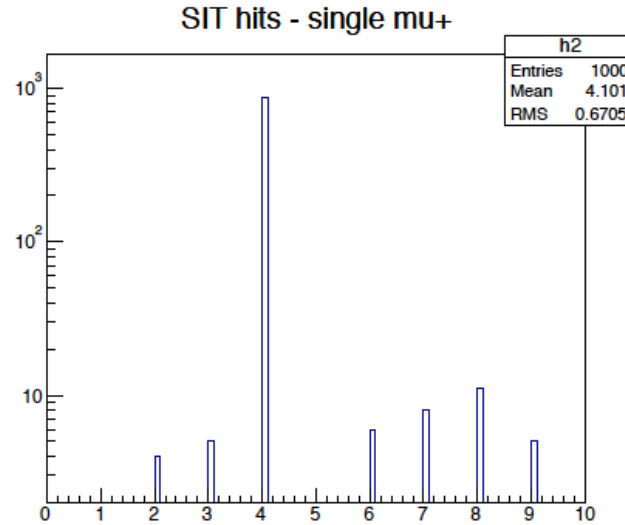
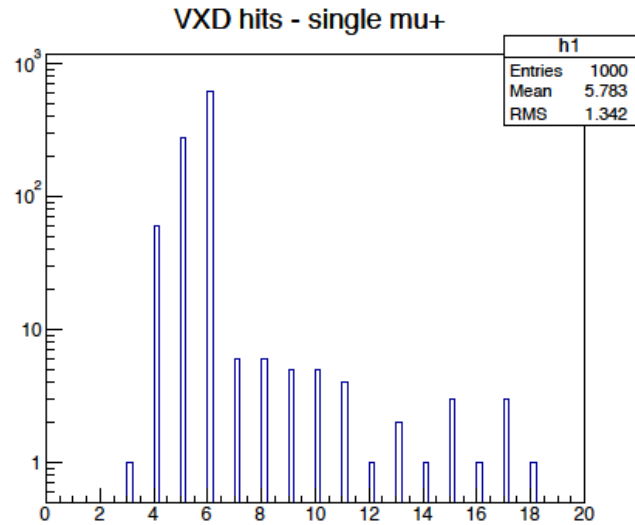
10 GeV Mu Hit Energy deposition in CLIC_SiD Si Barrel Tracker Modules



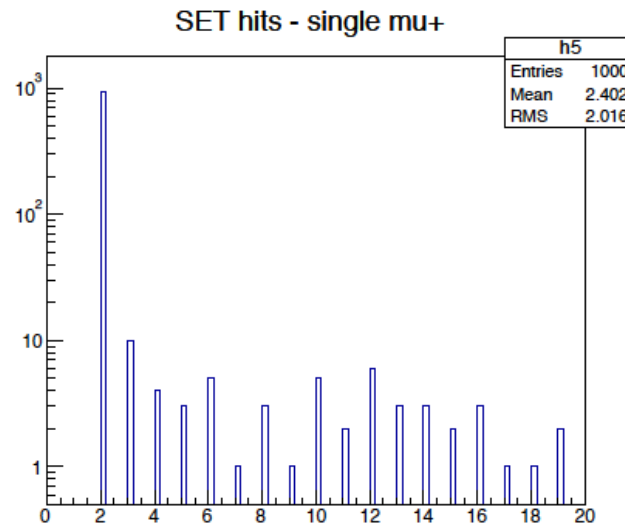
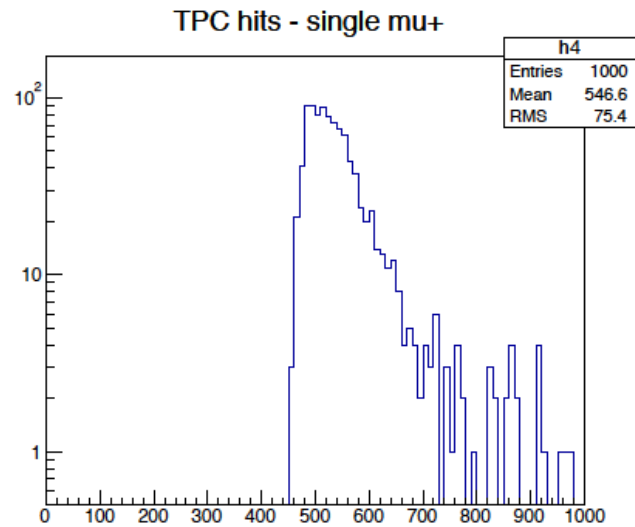
energy scale wrong by GeV/MeV: fixed !

checking hits in new model III

Frank Gaede, ILD SW&Ana Meeting, Nov 5, 2014

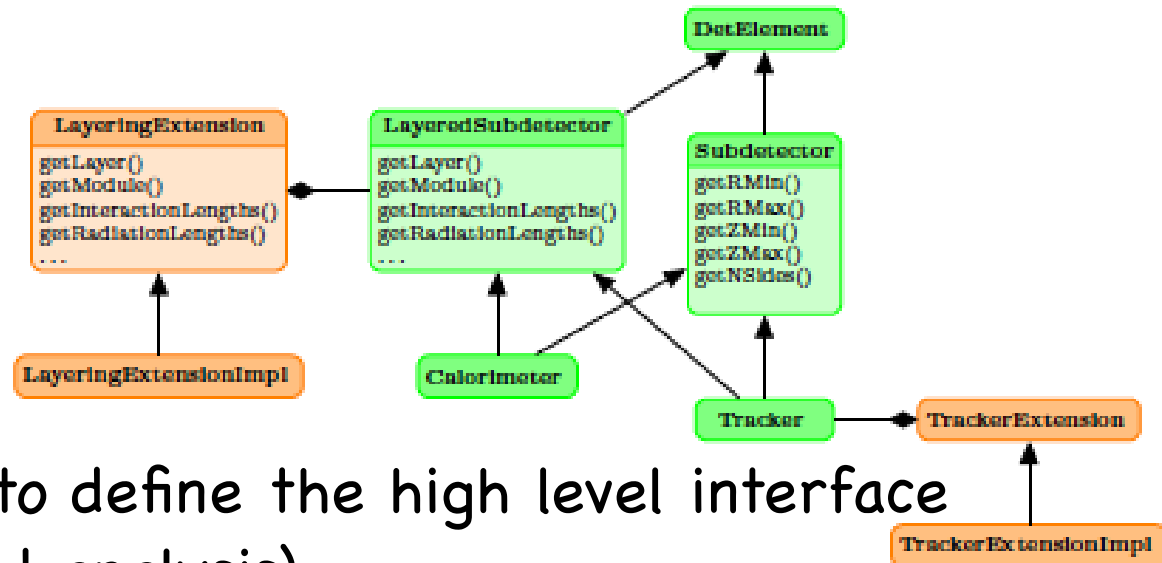


- sensitive detector works somewhat Ok for silicon trackers
- but poorly for the TPC



- development needed:
 - re-implement algorithms used in Mokka for Si and TPC
 - then could start serious validation of the model

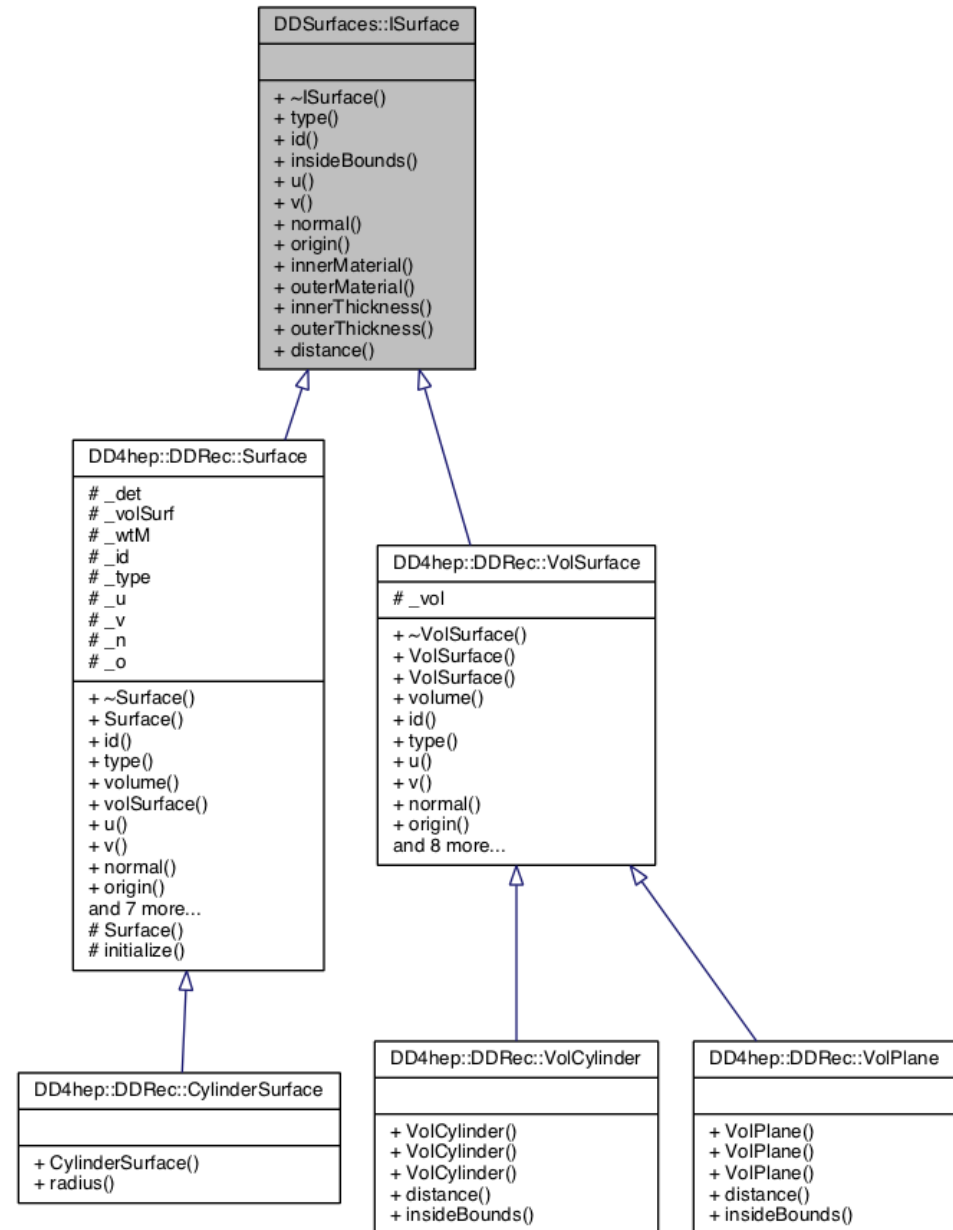
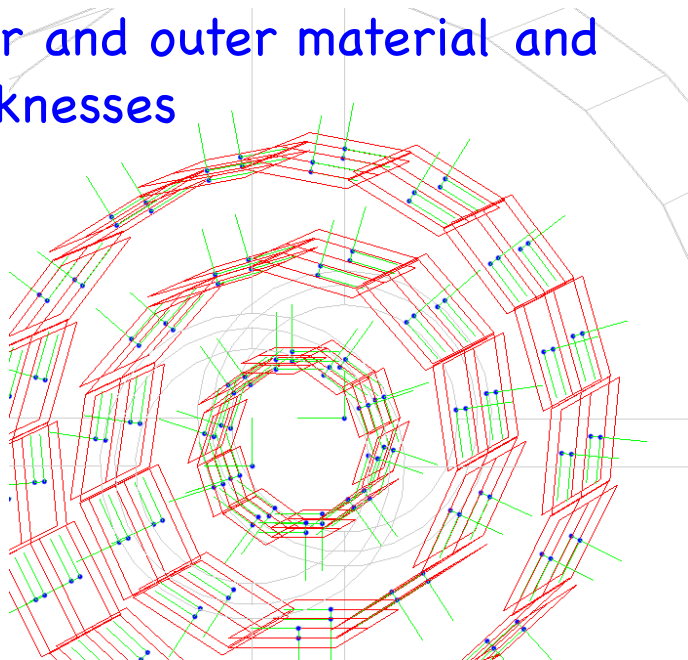
DDRec - interface for reconstruction



- use extension mechanism to define the high level interface for the reconstruction (and analysis)
- **calorimeters** are defined in hierarchy of **LayeredSubdetector** classes
 - provides access to layering structure and material properties per layer:
 - thickness, radiation and interaction lengths, cell sizes, ...
 - currently the API is refactored ...
- **trackers** are additionally defined through their measurement **surfaces** (+dead material surfaces)

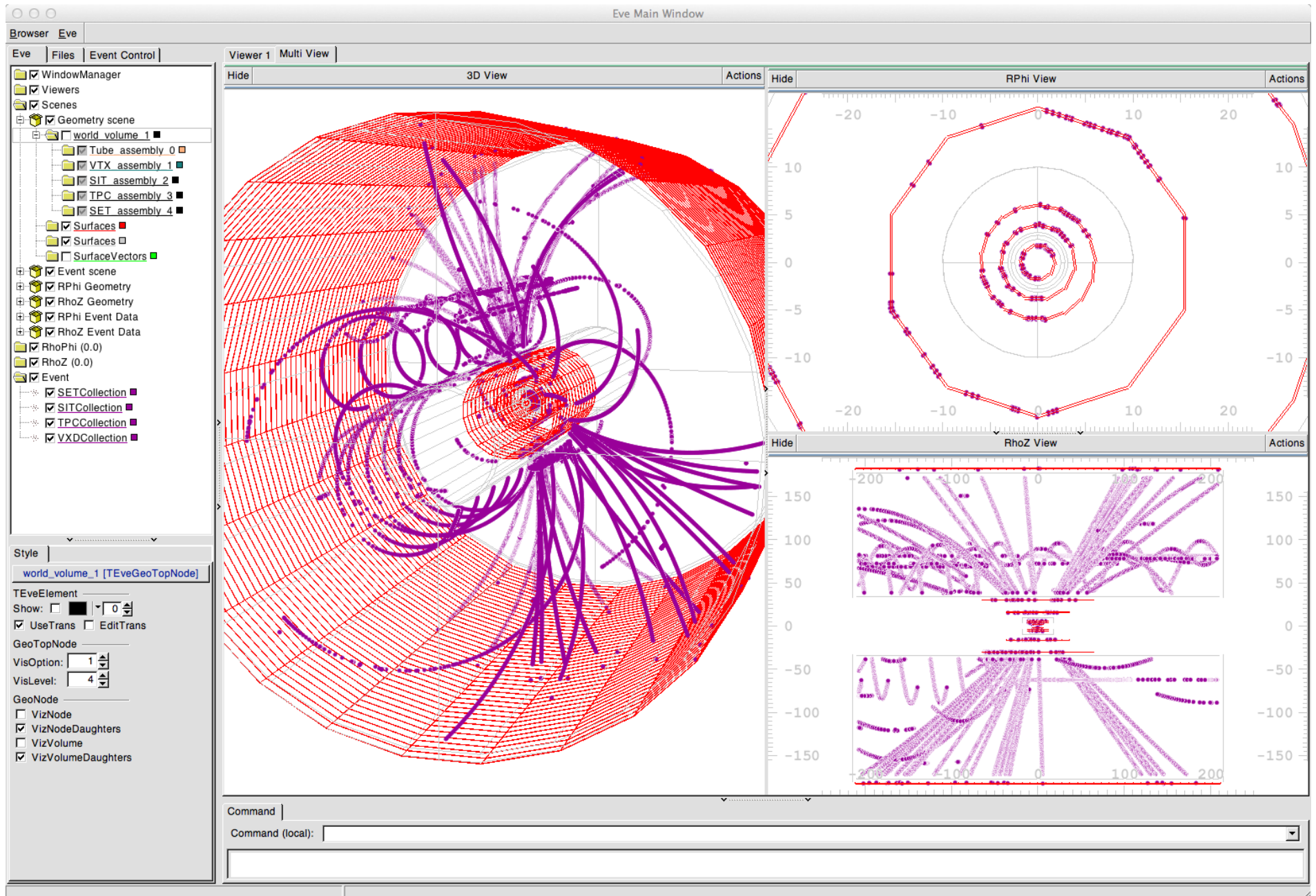
DDRec surfaces for tracking

- tracking code needs a special interface to geometry:
- measurement and dead material surfaces (planar, cylindrical)
- surfaces are attached to volumes (defining boundaries) and DetElements and provide:
 - u,v, normal, origin
 - inner and outer material and thicknesses



surfaces and LCIO hits in teveDisplay

Frank Gaede, ILD SW&Ana Meeting, Nov 5, 2014



DDRec versus GEAR

- need to be able to run existing reconstruction code **MarlinReco**, **MarlinTrk**, **PandaraPFA** and **LCFIPlus** with new simulation model
- completely based on **GEAR**
- -> need for (intermediate) interface from **DDRec** to **GEAR**

- current thinking:
- introduce light weight classes that hold relevant data and fill them on detector construction
- write 'quick and dirty' throw away code to generate GEAR objects from DD4hep detector model
- test and run existing reconstruction
- further develop new DDRec classes and eventually
- replace GEAR w/ DDRec in Marlin based reconstruction

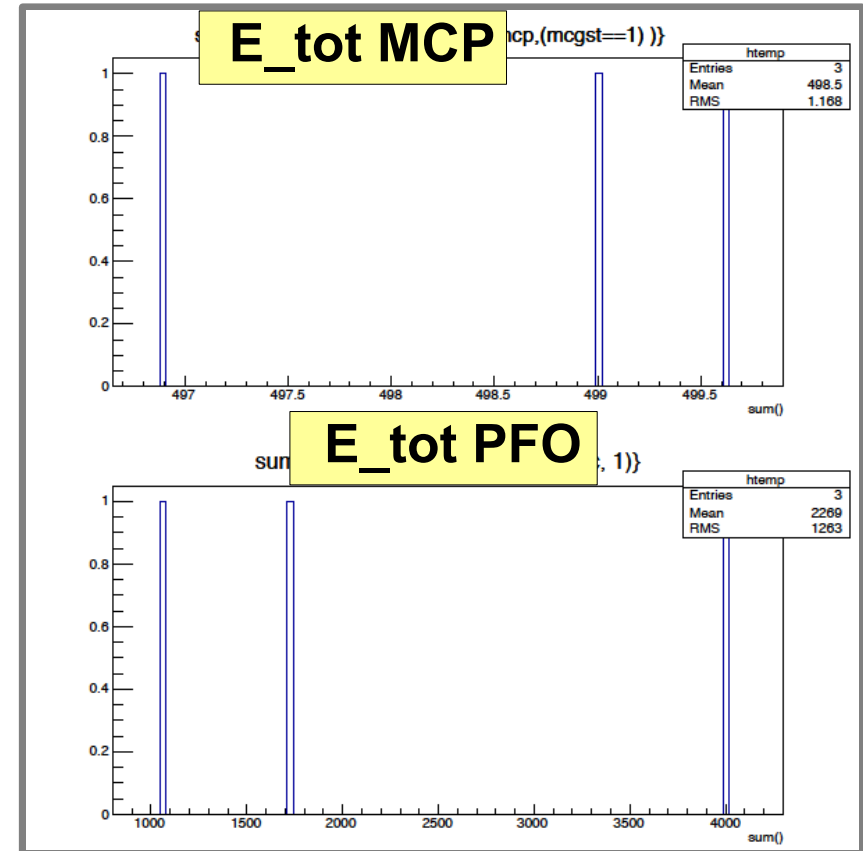
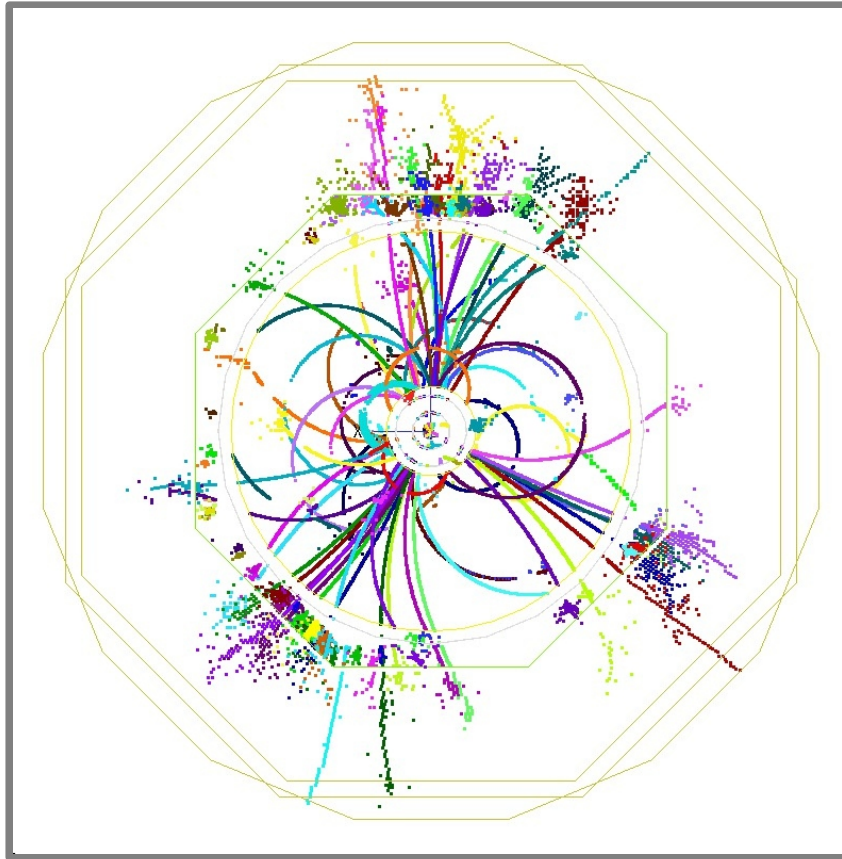
DDGear

- created lightweight data structures in DDRec to hold the relevant information that is used to instantiate Gear objects and/or create a Gear file:
 - FixedPadSizeTPCData - TPC
 - ZPlanarData - VXD, SIT, SET
 - ZDiskPetalsData - FTD
 - ConicalSupportData - beam pipe
 - LayeredCalorimeter - HcalBarrel, (HcalEndcap, Ecal , Lcal, Beamcal, LHcal, ... to be done)
- classes should be generic enough to serve also for other generic HEP detectors
- list could be extended

Running MarlinReco et al

- with the gear file created from DDSim model one can try and run the existing standard reconstruction
 - ILDConfig/StandardConfig/current/bbudsc_3evt_stdreco.xml
- a few adaptations are needed however:
 - [Clupatra/FixCellIDs](#):
 - changed to fix cellIDs also for SimTrackerHits
 - used for VXDCollection (DDSim: side=+-1 , Mokka side=0)
 - [MarlinTrkProcessor/SplitCollectionByLayer](#)
 - new processor - used to split up FTDCollection in
 - FTDPixelCollection and FTDStripCollection
 - [MarlinReco/NewLDCCaloDigi, SimpleLCalDigi, SimpleMuonDigi](#):
 - additional parameters: CellIDLayerString, CellIDModuleString, CellIDStaveString
 - deal with "K/K-1", "S-1", "M" to "layer", "stave", "module"
 - [MarlinPandora/src/CaloHitCreator.cc](#)
 - deal with "K/K-1", "S-1", "M" to "layer", "stave", "module"
- Note: all changes could also be done in additional processor to run old released versions of the code !

running the standard reco



- using DDGear* and the small adaptations one can run the ILD DBD standard reconstruction on the 'canonical' 3 ttbar events
 - events look nice at quick glance, but e.g. the energy scale (calibration) is wrong and probably many more things
- nevertheless this is a proof of concept
- need to finalize before serious testing starts

* using old Gear xml for calorimeters other than HcalBarrel

Summary & Outlook

- **DD4hep** now fairly complete
- **“DDSim”** - new detector description and simulation package
 - has first complete simulation model ILD_o1_v05 ported from Mokka
 - now also includes partly the services
 - started testing on hit level using the two sensitive detectors
- **DDRec** interface to reconstruction
 - added DDGear: conversion to Gear (files)
 - used to run standard ILD DBD reconstruction
- **Outlook:**
 - implement more sophisticated sensitive detectors
 - finalize interface to Gear
 - should be able to **start serious validation** of new model by ILD soon
- Xmas wish:
 - have a fully running version of the complete chain by end of year