Reconstruction and Core Software

Status and Plans

F. Gaede

CERN/DESY

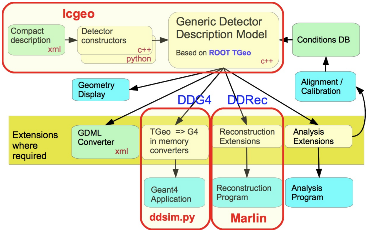
CLICdp Collaboration Meeting
CERN, June 2-3, 2015

Outline

1. **Core Tools**
   - DD4hep and lcgeo
   - Marlin and DD4hep
2. **DDRec**
   - DDRec detector description
   - Surfaces for Tracking
3. **Track reconstruction**
   - DDKalTest and IMarlinTrk
   - first results
4. **Calorimeter Reconstruction - PFA**
5. **Summary & Outlook**

Reminder: DD4Hep and lcgeo

- DD4hep provides detector geometry description for
  - simulation - via DDG4
  - reconstruction - via DDRec
- lcgeo is the common LC detector description package for ILD and CLIC
- simply python script for configuring and running the simulation: ddsim.py
- some common (CLICdp/ILD) geometry constructors for sub-detectors,
  - e.g. beamcal, ECal, Hcal,...
- first version of CLICdp simulation model exists (talk M.Petric)



lcgeo

Compact description xml

Detector constructors c++ python

Generic Detector Description Model

Based on ROOT TGeo c++

Conditions DB

Alignment / Calibration

Geometry Display

DDG4

DDRec

Extensions where required

GDML Converter xml

TGeo => G4 in memory converters

Reconstruction Extensions

Analysis Extensions

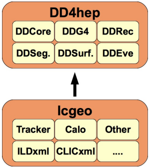
Geant4 Application

ddsim.py

Reconstruction Program

Marlin

Analysis Program



DD4hep

DDCore DDG4 DDRec

DDSeg. DDSurf. DDEve

lcgeo

Tracker Calo Other

ILDxml CLICxml ....

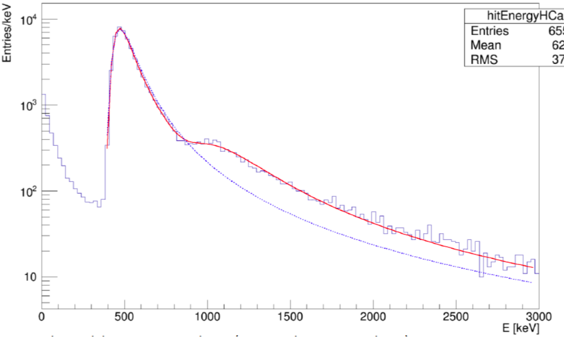
DD4hep - recent developments

- about to prepare new DD4hep release (v00-12)
- many changes since las release v00-11, e.g.
  - DDRec interface to eventually replace GEAR
  - SurfaceManager to access surfaces as needed for tracking
  - updated to Geant4 10.x series (requires at least Geant4 9.6)
  - started preparation for ROOT 6 (next release !?)
  - updated to optionally use C++11
  - introduced component structure: only link against what you use
  - introduced functionality for nested detectors and envelopes
  - added Birk's law for scintillator calorimeters
  - lots of fixes and improvements ...

- DD4hep is by now mostly feature complete for running Linear Collider simulation and reconstruction
- plan to have mostly bug fix releases for the near term future

Validation of DD4hep/lcgeo simulation

- Example:
- MIP hit energies in CLIC HCal (N.Nikiforou)
    - (10 GeV mu^-, uniform in ϕ)
- distribution consistent with two Landau distributions: one from the muon and a secondary from delta electrons



- more validation to come once model is ready...

Marlin and DD4hep

- currently Marlin depends on EDM and geometry, i.e. LCIO and GEAR
- try to avoid an additional dependency on DD4hep:
- created small standalone Marlin package: MarlinDD4hep
- this allows to have Marlin packages that do not need the geometry to not have to link against DD4hep

Note:

Every Marlin application that uses DD4hep, needs to run the InitializeDD4hep processor as first processor in the steering file !

DDRec detector description

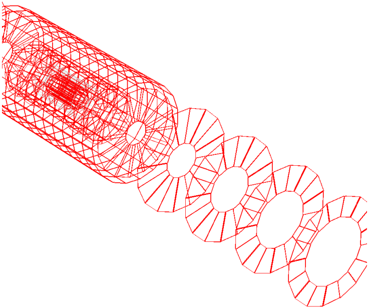
- simple data structures with high level view for reconstruction
  - attached to `DetElements`
  - similar to `GEAR` parameter classes

Data Structure	Detector Type	Example
`ConicalSupportData`	Cones and Tubes	BeamPipe
`FixedPadSizeTPCData`	Cylindrical TPC	TPC
`LayeredCalorimeterData`	Sandwich Calorimeters	ECal, HCal, fwd Calos
`ZPlanarData`	Planar Silicon Trackers	VXD, SIT, SET
`ZDiskPetalsData`	Forward Silicon Trackers	FTD

- can run exisiting (Gear based) Marlin processors on new simulation models: digitization, tracking, PFA,...
- eventually these data structs should (will) replace Gear

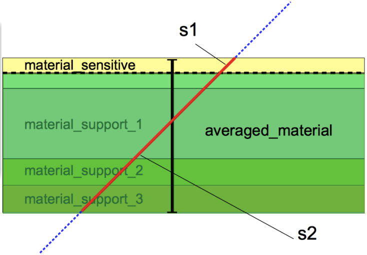
DDRec - tracking surfaces

- DDRec provides tracking surfaces (attached to volumes)
  - u,v, normal, origin
  - coordinate transforms: $(u, v) \leftrightarrow (x, y, z)$
  - material effects

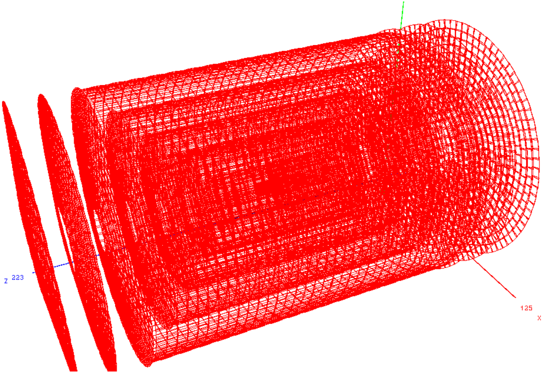


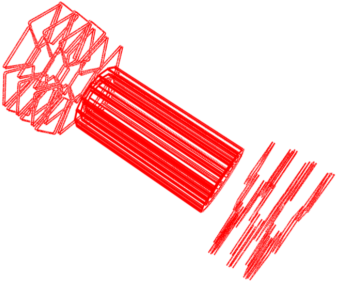
for multiple scattering and energy loss

- use averaged properties A, Z, ρ, X_0, λ
- with path lengths $s_{1,2} = thickness_{inner,outer} / cos(\alpha)$



tracking surfaces in CLIC model





Challenges for tracking

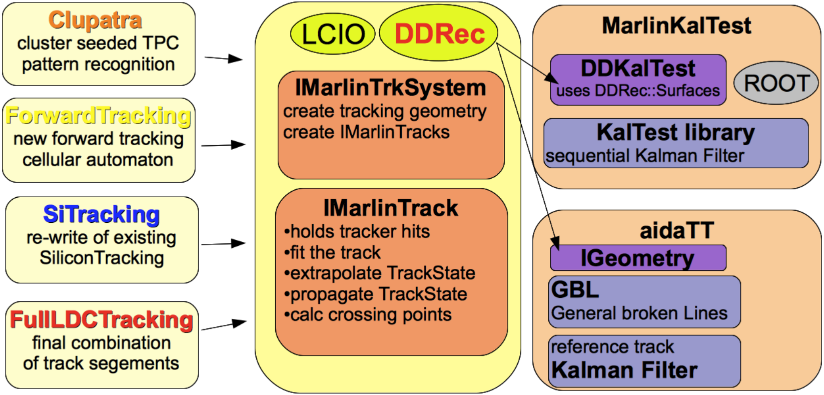
- large number of surfaces (> 19000): need efficient navigation
- spiraling vertex endcap: need new ideas for pattern recognition

DDKalTest

- DDKalTest provides generic interface between DD4hep and KalTest
  - uses `DDRec::Surfaces` instead of GEAR file
- DD(Parallel)PlanarMeasLayer
  - planar measurement layers (parallel and orthogonal to z)
  - works for 1D and 2D hits: VXD, SIT, SET, FTD, (all-silicon-tracker)
- DDCylinderMeasLayer
  - cylindrical measurement layers parallel to z: TPC

- with DDKalTestcan run the KalTest fitter on any tracking detector that has the `DDRec::Surfaces` implemented
- no additional glue code needed!

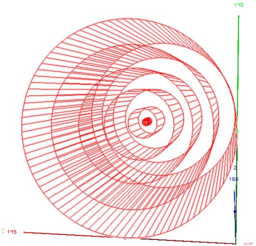
IMarlinTrk Interface and DDRec



- pattern recognition separated from fitting via IMarlinTrk
- currently still use GEAR for (little) geometry information for patrec
  - rather straight forward to replace w/ DDRec DetectorData classes
- updated all tracking processor to allow for chooosing the fitting type:
  - SiliconTracking, ForwardTracking, Clupatra, FullLDCTracking, Refitting, ...

Track fitting in an all Si-Tracker

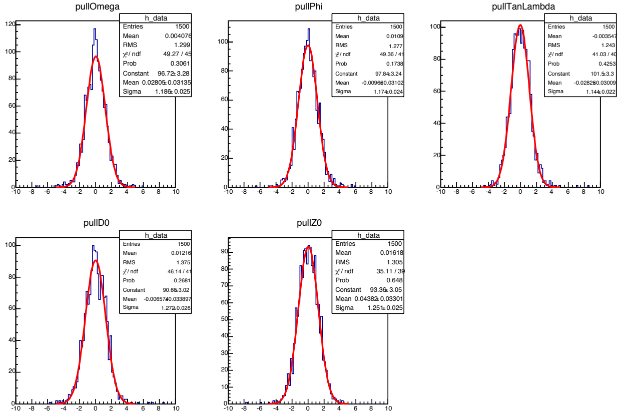
- expected track fitter to also work for all Si-tracker of CLIC out of the box
  - however many single μ-track fits failed and pulls were observed to be not correct
- used simplified tracking model with a large Si-strip barrel tracker to develop new



Fitting strategy for CLIC tracker

- 1D hits provide no constraint in z and thus cannot be used to initialize track parameters
- need to fit inside-out - starting with vertex pixel hits
- finally smooth back to third hit and fit inside from there

Track fitting in new CLIC model



- observe reasonably nice looking pull distributions for track parameters in the detailed CLIC tracker model (R.Simoniello)
- (single mu^{-}, 5 GeV at $\theta = 85\,\mathrm{deg}$)

- fairly confident that new fitting code using the DDRec::Surface and material averaging works
- will focus on track finding from now on ...

Strategies for Pattern Recognition

option 1

- implement a track cheater (MCTruth) to decouple from the development of the rest of reconstruction chain (PFA)

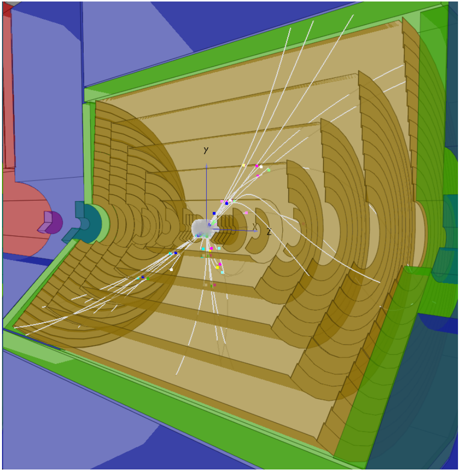
option 2

- continue development of ILD CA-based Vertex pattern recognition for finding seed tracks followed by extrapolation outwards to Silicon Barrel Tracker (R.Simoniello)
- try and apply (adapt) ILD ForwardTracking to endcaps

option 3

- develop a CLIC specific pattern recognition by adopting and modifying the CA based pattrec to the specific layout of the new CLIC detector model
- investigate other ideas (conformal mapping, Hough Transform, etc...)

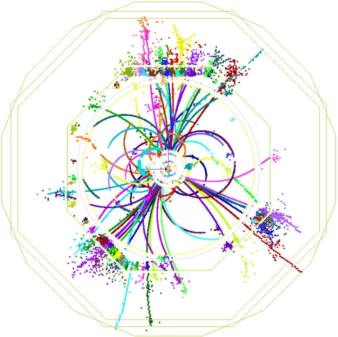
First look at real events



- can find more than one track with option 2 in hadronic Z-event (R.Simoniello)
- time to start now to seriously develop, test and debug the pattern recognition

Calorimeter Reconstruction - PandoraPFA

- strategy for calorimeter reconstruction:
- adopt MarlinPandora to use DDRec::LayeredCalorimeter instead of GEAR
- use cheated tracks as input to Pandora
- apply calibration procedure for PandoraPFA
- ⇒ fairly straigh forward (but of course work)
- ⇒ time to start now



example $t\bar{t}$ event in ILD simulated w/ DD4hep and reconstructed in Marlin

Summary & Outlook

- core software tools DD4hep and lcgeo are essentially feature complete
- DDRec interface to reconstruction now also finalized
- track fitting using DDRec::Surfaces demonstrated to work in CLIC barrel tracker

Next major steps

- increase the efforts on development of the pattern recognition
- work on getting PandoraPFA to work new simulation model

Outlook - Goal

- have a running version of the simulation and reconstruction running this summer:
- ambitious but feasible