Forward Calorimeter Simulation and Reconstruction with DD4hep

André Sailer

CERN-PH-LCD

FCal Workshop
March 23, 2015

Table of Contents

1 DD4HEP
- BeamCal
- LumiCal
- Beam-Pipe and Masks

2 Simulation
- Segmentations in DD4HEP

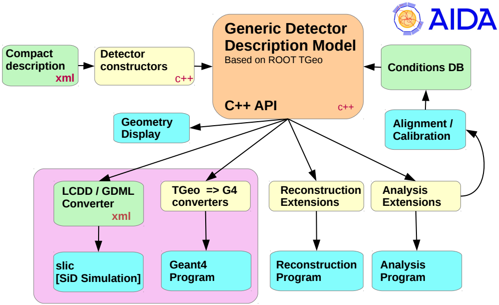
3 Reconstruction

4 Summary

DD4HEP

DD4HEP: Detector description for HEP

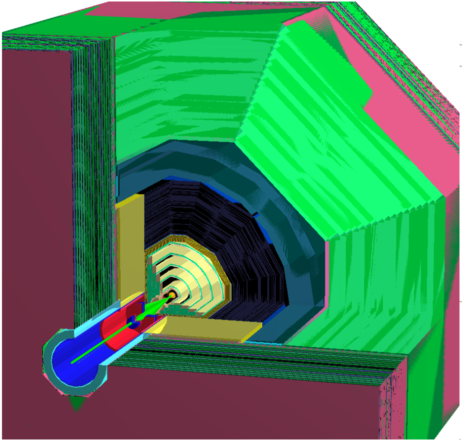
- Geometry description system to be single source of geometry for simulation, reconstruction, analysis
- Simulation with Geant4: ‘DDG4’
- Interfaces for reconstruction
  - Segmentations to convert positions to cell IDs and back: ‘DDSegmentation’
  - Alignment
- Linear collider detector geometry combined in the ‘LCGEO’ package

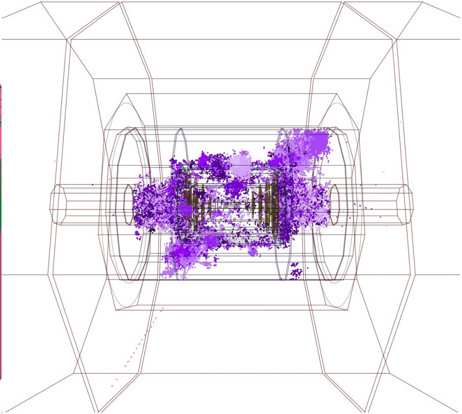


M. Frank

New Detector Models

- CLICdp and ILD will base their future full detector simulations on DD4HEP
- Replacing Mokka, and adding functionality required for reconstruction





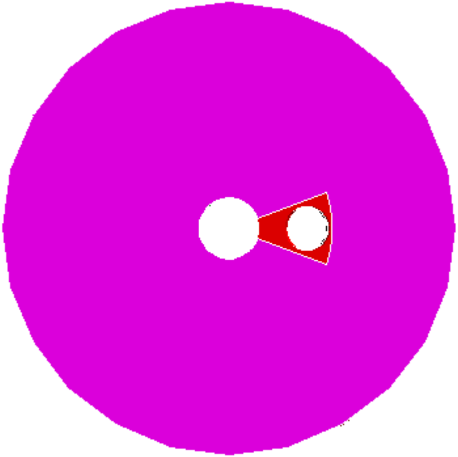
(F.Gaede)

Very Brief Installation Guide

- Very short instructions to compile DD4HEP and LCGEO can be found here: https://twiki.cern.ch/twiki/bin/view/CLIC/CLICDD4hep
- Easier to keep up-to-date than slides, because these things are still subject to change

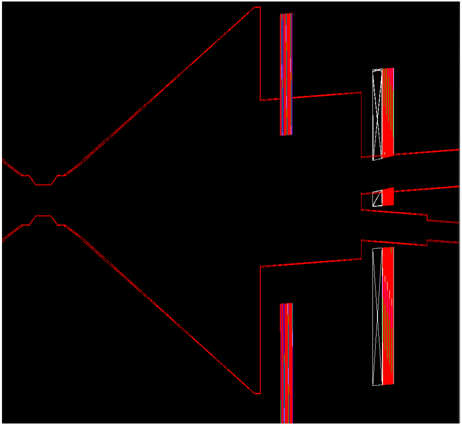
BeamCal Description Update

- BeamCal implemented for some time
  - Only holes in the absorber (red)
  - Keyhole shape for the sensitive parts (magenta)
- Discovered issues with rotation and location of cutouts for incoming/outgoing beam pipe
  - Incoming and outgoing beam pipe on the wrong side
  - Rotation on the wrong side



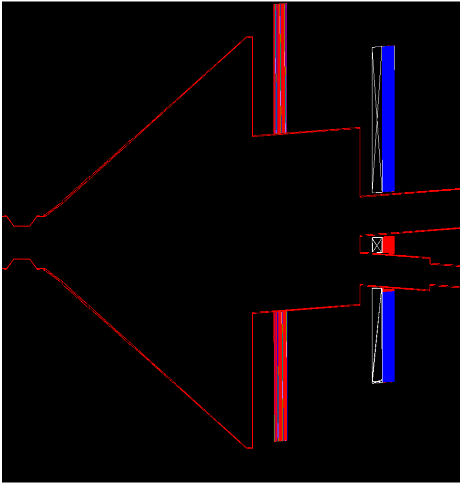
BeamCal Description Update

- BeamCal implemented for some time
  - Only holes in the absorber (red)
  - Keyhole shape for the sensitive parts (magenta)
- Discovered issues with rotation and location of cutouts for incoming/outgoing beam pipe
  - Incoming and outgoing beam pipe on the wrong side
  - Rotation on the wrong side



BeamCal Description Update

- BeamCal implemented for some time
  - Only holes in the absorber (red)
  - Keyhole shape for the sensitive parts (magenta)
- Discovered issues with rotation and location of cutouts for incoming/outgoing beam pipe
  - Incoming and outgoing beam pipe on the wrong side
  - Rotation on the wrong side



BeamCal XML

BeamCal composition can be changed via the XML

- Number of layers, thicknesses, materials
- Different thicknesses for different layers

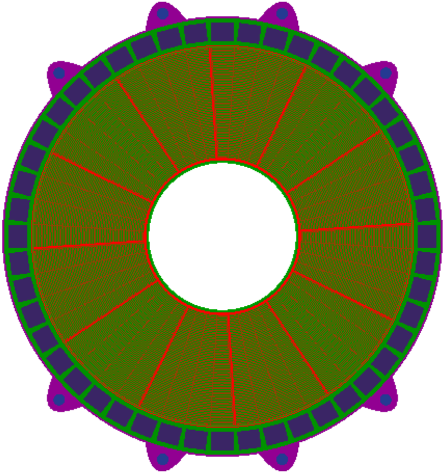
```xml
<detector name="BeamCal" type="BeamCal_o1_v01" vis="BeamCalVis" id="14"
          readout="BeamCalHits">
  <parameter crossingangle="CrossingAngle"
             cutoutspanningangle="360*degree-BCal_SpanningPhi"
             incomingbeampiperadius="BCal_TubeIncomingRadius"
             />
  <dimensions inner_r="BCal_rInner" inner_z="BCal_zInner" outer_r="BCal_rOuter" />
  <layer repeat="1">
    <slice material="C" thickness="BCal_dGraphite"
           layerType="holeForIncomingBeampipe"/>
  </layer>
  <layer repeat="BCal_nLayers">
    <slice layerType="holeForIncomingBeampipe"
           material="TungstenDens24" thickness="BCal_dAbsorber" />
    <slice material="Silicon"         thickness="0.3004*mm" sensitive="yes"  />
    <slice material="Copper"          thickness="0.0004*mm" />
    <slice material="Kapton"          thickness="0.15*mm"   />
    <slice material="Air"             thickness="0.05*mm"   />
  </layer>
</detector>
```

LumiCal (M. Petric)

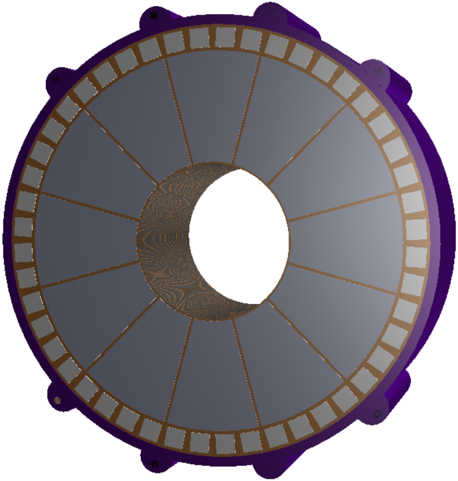
LumiCal description

- Simplified model existing, very similar to BeamCal
- More detailed model still needs finalisation

MOKKA

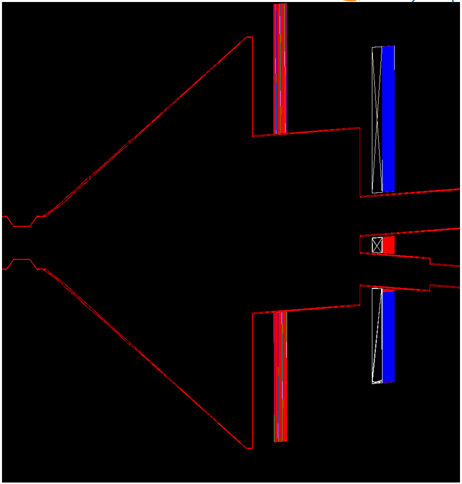


DD4HEP



Beam-Pipe and Masks

- Converted Mokka beam-pipe (F.Gaede) and mask driver to LCGEO
- Place “Cylinders” at arbitrary location
- Allow for rotation and cut-outs
- Beam-pipe allows more shapes to create a ‘closed’ vacuum system
  - slices ends off rotated cylinders to make their ends perpendicular to z-axis

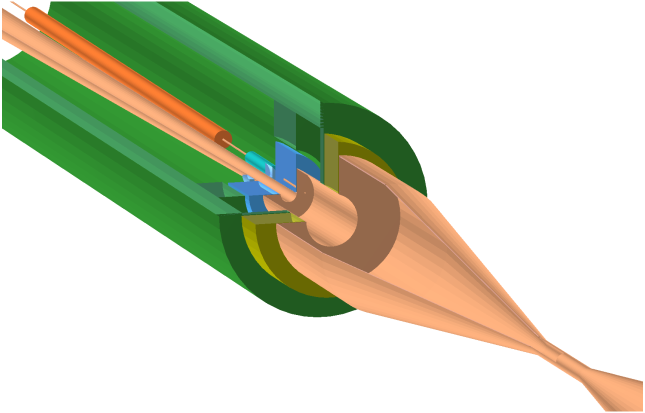


Example description:

```
<detector name="Beampipe" type="Beampipe_o1_v01" >
  <parameter crossingangle="CrossingAngle" />
 <section type="Center" start="0*mm" end="260*mm" rMin1="29.4*mm" rMin2="29.4*mm"
            rMax1="30.0*mm" rMax2="30.0*mm" material="Beryllium" name="BP01"/>
</detector>
```

Full Forward Region

Description of the forward region is complete:



Simulation

- Now that the geometry is described, start thinking about simulation
- Requires a *Sensitive Detector* and a *segmentation* of the sensitive area

Sensitive Detector:

- Treats the deposited energy in the sensitive volume
- Can use a generic treatment and simply take the deposited energy without conversions at this step

Segmentation

- Segmentation of the sensors; which energy deposits to merge into combined hits

In Mokka sensitive detectors and segmentations were closely merged; segmentation needed to be duplicated for the reconstruction

Segmentations in DD4HEP

- Segmentations are a standalone package and can be used separately of DD4HEP
- Different segmentations can be mixed with the sensitive detectors
- Any segmentation currently foreseen for the forward calorimeters can be described:
  - Uniform Segmentations with constant R/Phi, R/RPhi or X/Y sizes
  - Arbitrary R and RPhi sizes (e.g.: Lucia's proportional segmentation)
- Different segmentations can be easily exchanged
  - R-Phi and X-Y segmentations are implemented, and cell sizes can be varied freely
  - Almost automatic scaling is also possible via the XML file
- New segmentations can be added as plug-ins

Uniform Segmentations

Constant grid sizes

- XY-Segmentation: (SLIC BeamCal currently uses this):

```
<segmentation type="CartesianGridXY" grid_size_x="3.5*mm" grid_size_y="3.5*mm" />
```

- Constant R-Phi, e.g. foreseen for the LumiCal:

```
<segmentation type="PolarGridRPhi"
              grid_size_r="1.8*mm" grid_size_phi="360/64*degree"
              offset_r="InnerRadius" />
```

Non-Uniform Segmentations

E.g.: Constant steps in R and almost constant size in RPhi (‘traditional’ BeamCal segmentation)

```
<segmentation type="PolarGridRPhi2"
              grid_r_values="3.200*cm  3.9876*cm 4.7742*cm 5.5608*cm
                             6.3474*cm 7.134*cm 7.9206*cm 8.7072*cm
                             9.4938*cm 10.2804*cm 11.067*cm 11.8536*cm
                             12.6402*cm 13.4268*cm 14.2134*cm 15.0*cm"

              grid_phi_values="BCal_SpanningPhi/(4*8)*deg
                               BCal_SpanningPhi/(5*8)*deg
                               BCal_SpanningPhi/(6*8)*deg
                               ...
                               BCal_SpanningPhi/(15*8)*deg"

              offset_phi="-180*deg+(360*deg-BCal_SpanningPhi)*0.5"
              />
```

(“...” just to save space, not automatic repetition)

Non-Uniform Segmentations

E.g: Proportional segmentation: radius steps become larger (*BCalProF* $\approx$ 1.1), exponent operator ^ is parsed as expected, constant size in Phi

```
<segmentation  type="PolarGridRPhi2"
               grid_r_values="BCal_rInner
                              BCal_rInner*BCal_ProF^1
                              BCal_rInner*BCal_ProF^2
                              ...
                              BCal_rInner*BCal_ProF^14
                              BCal_rOuter"

               grid_phi_values="360*deg/63  360*deg/63  360*deg/63
                                ...
                                360*deg/63"

               offset_phi="-180*deg+(360*deg-BCal_SpanningPhi)*0.5"
               />
```

(“...” just to save space, not automatic repetition)

Simulation with DD4HEP

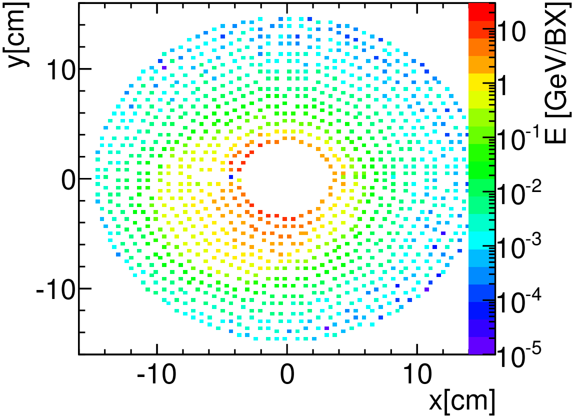
- Simulation is done with GEANT4
  - Chose physics list, input files, Lorentz boost, . . .
- Almost all features implemented
  - Still missing: magnetic field maps (e.g., for Anti-DID).
- Can configure simulation parameters via `python`, XML, or CINT
  - But this needs some more polishing

Simulated a single bunch crossing of CLIC 3 TeV incoherent pairs in the BeamCal with different segmentations

Different Segmentations

- Not showing exact segmentation, just filled into 2D Histogram
- Floating point precision smears out the boundary

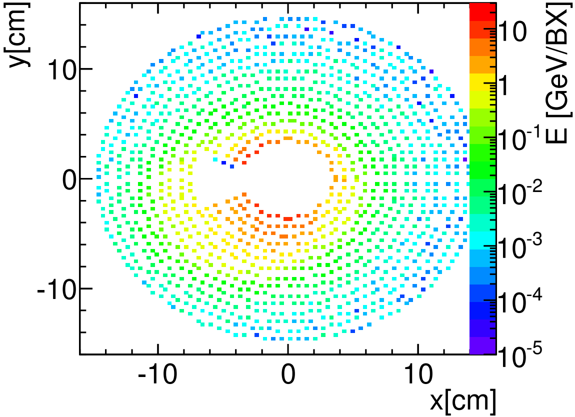
CLIC-like: Small cutout angle:



Different Segmentations

- Not showing exact segmentation, just filled into 2D Histogram
- Floating point precision smears out the boundary

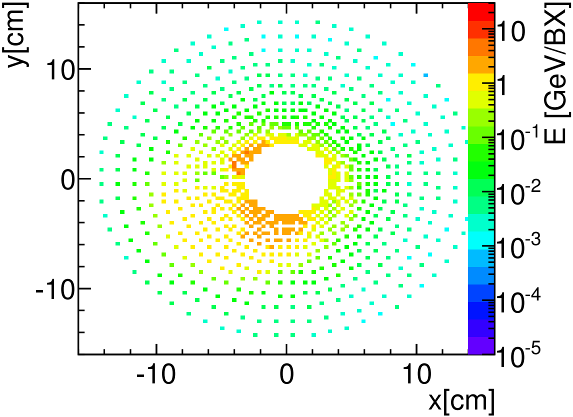
ILD-Like: Large cutout angle:



Different Segmentations

- Not showing exact segmentation, just filled into 2D Histogram
- Floating point precision smears out the boundary

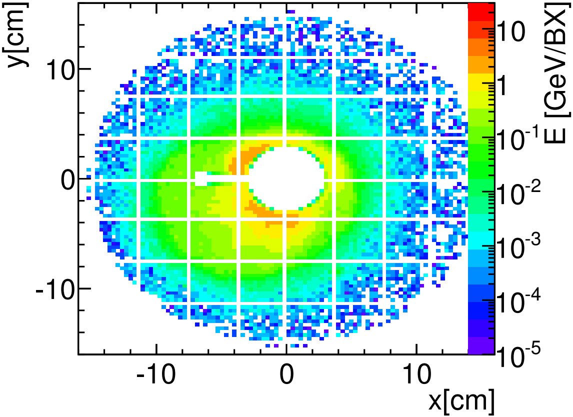
Proportional with small cutout angle:



Different Segmentations

- Not showing exact segmentation, just filled into 2D Histogram
- Floating point precision smears out the boundary

XY-Segmentation:



Reconstruction

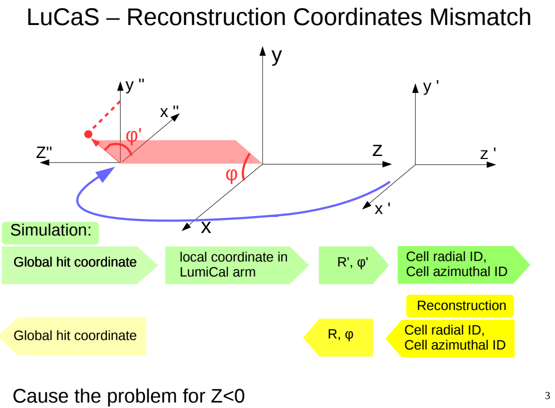
- Reconstruction in FCalClusterer package rely on cell IDs to identify cells and create clusters
  - Cells are neighbours if their IDs differ by 1
  - Calculate position based on cell IDs
- Transformation from cell IDs to local to global coordinates needs to be available

Cell IDs and Local Coordinates

- Direct access to the local coordinates via the segmentation
  `position = mySegmentation.position(cellID)`
  - Can obtain local coordinates from cell IDs by directly using the same code used in the simulation
  - Can greatly simplify reconstruction code and make it more generic
  - Remove assumptions on the segmentation used where-ever possible
- Direct access to coordinate transformations from the DD4HEP-Geometry

Local < – > Global Confusion

- Recent problem discovered by Sasha
- Local to global coordinate transformation not done correctly, global position is wrong
- With DD4HEP direct access to the transformation matrices
  - Calculate global position from local coordinates *for free*



LuCaS – Reconstruction Coordinates Mismatch

Simulation:

Global hit coordinate → local coordinate in LumiCal arm → R', φ' → Cell radial ID, Cell azimuthal ID

Reconstruction

Cell radial ID, Cell azimuthal ID → R, φ → Global hit coordinate

Cause the problem for Z<0

3

(S. Borysov)

```
LumiCal.child("LumiCalCal01").localToWorld(local,global);
LumiCal.child("LumiCalCal01").worldToLocal(global,local);
```

Summary

Simulation

- Simulation of Forward Calorimeters almost ready to be done in DD4HEP
- Minor issue with Segmentation and missing magnetic field-maps

Reconstruction

- Reconstruction can benefit from additional features provided by DD4hep
- Access to the geometric transformation
- No more code-duplication for segmentations
- Simplified code