

Forward Calorimeters in DD4hep

André Sailer, Marko Petric
CERN-PH-LCD

FCal Collaboration Workshop
Belgrade, Serbia
October 12–13, 2014

Content



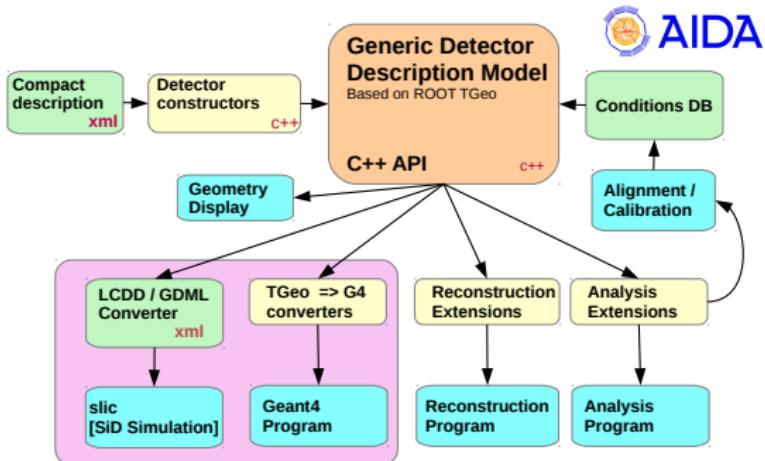
- 1 DD4hep
- 2 Detector Geometries
 - BeamCal
 - LumiCal
 - LHCAL
- 3 Geant4 Simulation of DD4hep Geometry
 - Segmentations
- 4 How to Install and Run Simulation with DD4hep
- 5 Summary

DD4hep

DD4hep provides

- Geometry description system
- Single source of geometry for simulation, reconstruction, analysis
- Simulation with Geant4: 'DDG4'
- Segmentations to convert positions to cell IDs and back: 'DDSegmentation'
- Interface for reconstruction software: 'DDRec'

For LumiCal and BeamCal, only the *Detector Constructor* and the XML file need to be written



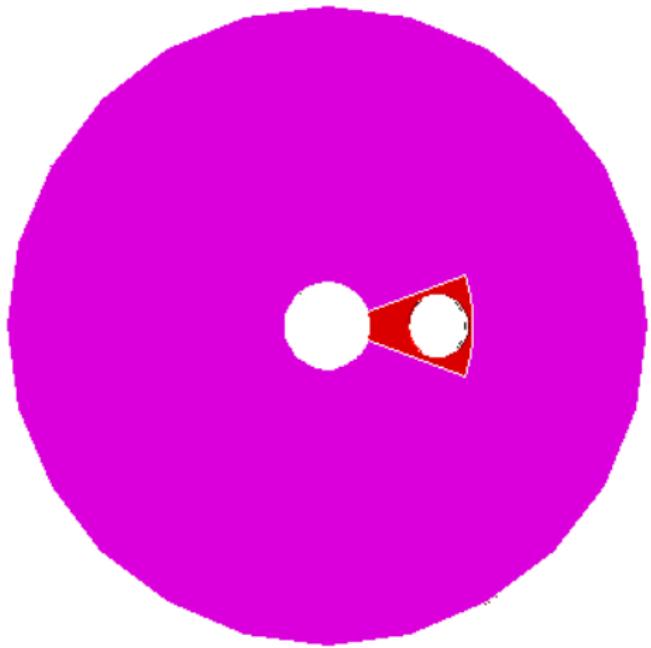
M. Frank

Detector Geometries



- On the next slides the current status and outlook of the forward calorimeter geometries is presented
- One example of the XML steering for the detector geometry will also be shown

- Re-written driver from Mokka in DD4hep
- More flexibility: layer sequence completely controllable by xml
- Can change absorber thickness, sensor thickness, from layer to layer
- Notice the difference between the absorber layer (red) and sensor layer (magenta)



BeamCal XML



```
<detector name="BeamCal" type="BeamCal" vis="SeeThrough" id="ILDDetID_BCAL"
          readout="BeamCalCollection" insideTrackingVolume="false" >

<parameter crossingangle="ILC_Main_Crossing_Angle"
           cutoutspanningangle="40*deg"
           incomingbeampiperadius="BCal_TubeIncomingRadius" />

<dimensions inner_r = "BCal_rInner"
             inner_z = "LHcal_zend_+LHcal_BCal_clearance"
             outer_r = "BCal_rOuter" />

<layer repeat="1">
  <slice material = "C" thickness = "BCal_dGraphite"
        layerType="holeForIncomingBeampipe" />
</layer>

<layer repeat="BCal_nLayers">
  <slice material = "TungstenDens24" thickness = "BCal_dAbsorber"
        layerType="holeForIncomingBeampipe" />
  <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>
```

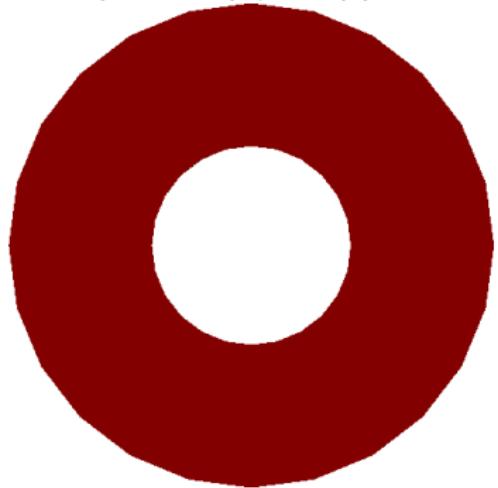
Description of XML Snippet



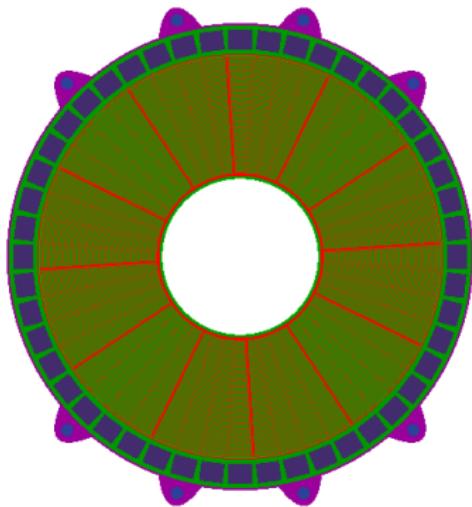
- Global parameters are passed to the detector, not directly taken from global namespace
- Radii and starting position are described via XML constants, scaling is done if at all at the XML level
- The layer structure is completely flexible
 - ▶ The graphite block and the absorber layers are described in the same way
 - ▶ Layer structure can be defined per slice
- The XML structure for LumiCal is very similar, except there are no different layer structures, yet
- Could also make thin layer of silicon for the PairMonitor
- Most visualisation attributes have been dropped, everything can have their own visualisation

- Only very basic geometry implemented at the moment
- Bogdan pointed us to his more detailed description only last week
- Will implement gaps between sensor wedges soon
- Maybe also implement the support structure surrounding the LumiCal

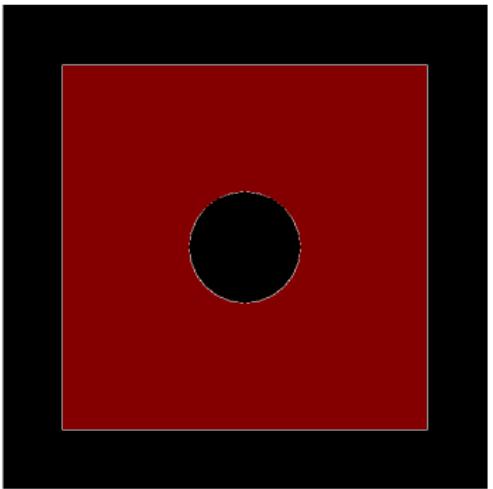
Geometry in dd4hep (dummy parameters)



Geometry in LuCaS



- Implemented as in Mokka



Geant4 Simulation of DD4hep Geometry



- DD4hep geometry written in TGeo can be translated to Geant4 geometry in memory
- DDG4 part of DD4hep
- DDG4 also allows steering of Geant4 physics list, input, output, etc.
 - ▶ Replacement for Mokka
- Sensitive detectors are factorised between treatment of energy and segmentation of sensors
- It is also possible to write the DD4hep geometry to GDML and then use that as an input for SLIC

Segmentations



- Segmentations convert between position of the hit and cell ID and back
- Segmentation and sensitive detector independent (not as in Mokka)
- Describe any kind of R/Phi segmentation (work in progress)

```
<readout name="BeamCalCollection">
  <segmentation type="CartesianGridXY"
    grid_size_x="3.5*mm"
    grid_size_y="3.5*mm"
    offset_x="0*cm"
    offset_y="0*cm" />
  <id>system:8 , barrel:3 , layer:5 , slice:5 , x:-16,y:-16 </id>
</readout>
<readout name="LumiCalCollection ">
  <segmentation type="PolarGridRPhi"
    grid_size_r="1.8*mm"
    grid_size_phi="130.899*mrad"
    offset_r="80*mm"   />
  <id>system:8 , barrel:3 , layer:5 , slice:5 , r: -16,phi: -16 </id>
</readout>
```

- Here two types: CartesianGridXY, PolarGridRPhi
- ID string to enumerate bits per field

How to Install



Requirements/Installation: cmake, Root (>5.34.10), Geant4, boost, LCIO

```
svn co https://svnsrv.desy.de/public/aidasoft/DD4hep/trunk
svn https://svnsrv.desy.de/public/ddsim/DDSim/trunk
cd dd4hep; mkdir build; cd build;
cmake -DDD4HEP_USE_BOOST=True -DDD4HEP_USE_GEANT4=True \
      -DDD4HEP_USE_LCIO=True \
      -DGeant4_DIR=$G4LIB -DCLHEP_DIR=$CLHEP_BASE_DIR \
      -DLCIO_DIR=$LCIO
make install; source $DD4HEP/bin/thisdd4hep.sh
cd ddsim; mkdir build; cd build; cmake ..;
make install; source $DDSIM/bin/thisddsim.sh
```

- Set variables to correct values: LCIO, G4LIB, CLHEP_BASE_DIR, DD4HEP, DDSIM
- Can also be installed via ilcsoft-install

- There are a few executables compiled via DD4hep/DDSim
- geoDisplay: Root TGeo geometry
 - ▶ geoDisplay compact.xml
- teveDisplay: event Root TEve event display
 - ▶ teveDisplay compact.xml
- dd_sim: Geant4 simulation, controlled via XML (or python or CINT)
 - ▶ dd_sim compact.xml sequence.xml physics.xml
 - ▶ Additionally need describe which sensitive detectors, physics list, input, output,
...
▶ cd DDSim/example;
dd_sim ..//ILD/compact/BeamCalOnly2.xml BCPhys.xml BCSeq.xml
- Examples in DD4hep or DDSim subfolder *examples*
- ILD compact XML files in DDSim/ILD/compact

Summary



- Forward calorimeters almost ready for full simulation
- Some details for precision studies still need to be implemented
- Need some testing