Detector Description in CEPCSW

Chengdong FU

(IHEP, CAS)

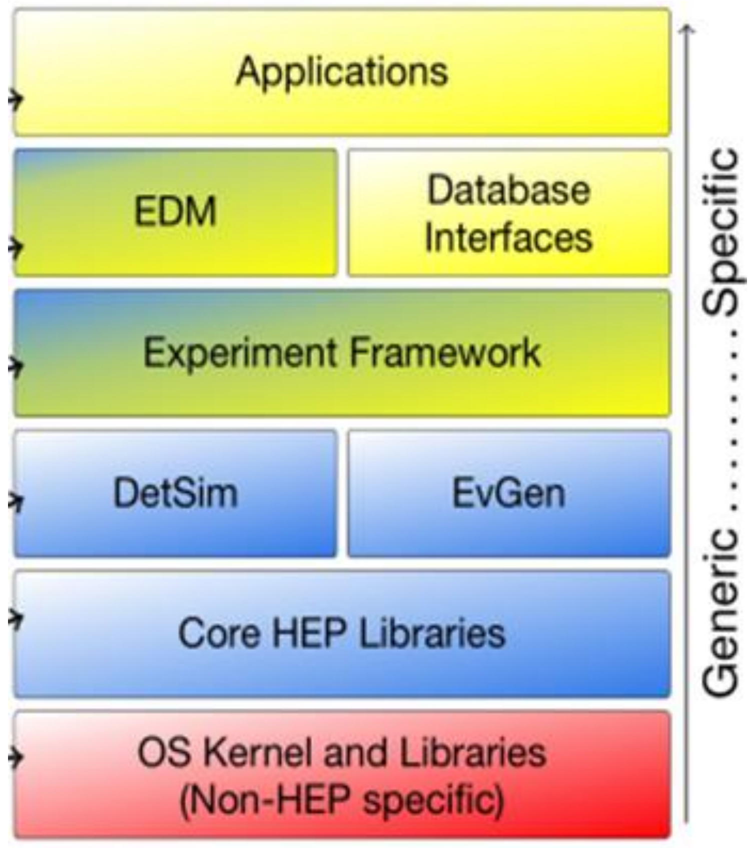
Workshop on Detector & Accelerator Mechanics

Dongguan, 2020-08-29

Contents

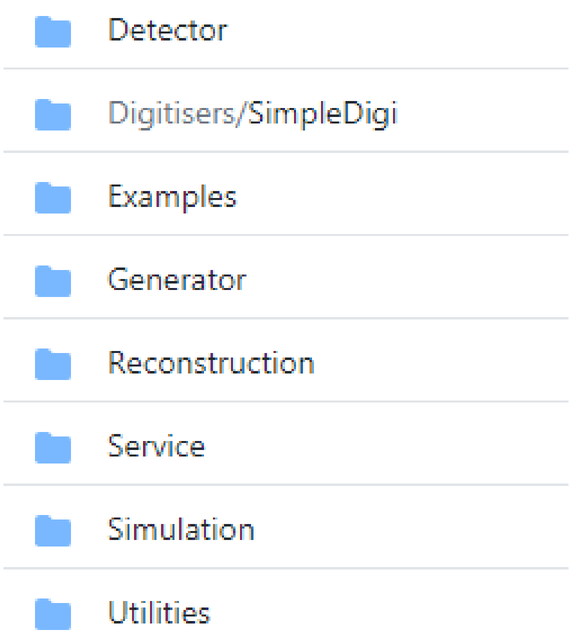
Introduction

- Key4hep
  - Geant4
  - ROOT
  - Gaudi
  - etc.
- CEPCSW
  - Gaudi
  - EDM4hep
  - Geant4
  - DD4hep for detector desription
  - ROOT
- Compare to CepC software used in CDR
  - LCIO → EDM4hep
  - MokkaC → DD4hep (DB→XML) [XML also can keep in DB]
  - Marlin → Gaudi
  - Gear → DD4hep



How to Start

- setup environment
  - source /cvmfs/cepcsw.ihep.ac.cn/prototype/releases/externals/97.0.2/setup.sh
- download packages from github
  - git clone http://github.com/cepc/CEPCSW.git
- compile & link
  - mkdir build & cd build
  - cmake .. -DHOST_BINARY_TAG=${BINARY_TAG}
  - make
- modify code or create new & make
  - Detector description:
    - /Detector/DetCEPCvx
    - compact/
    - src/
      - include/
      - calorimeter
      - tracker
      - other



Detector
Digitisers/SimpleDigi
Examples
Generator
Reconstruction
Service
Simulation
Utilities

Geant4 Construction

- Definition of material
  - User define:
    - G4Isotope → G4Element → G4Material
  - NIST database:
    - G4NistManager* manager = G4NistManager::Instance();
    - G4Material* air = manager->FindOrBuildMaterial(“G4_AIR”);
- Construct detector
  - G4RunManager* runManager = new G4RunManager;
  - XXXDetectorConstruction* detector = new XXXDetectorConstruction;

  - runManager->SetUserInitialization(detector);
  - runManager->SetUserInitialization(new XXXPhysicsList());

  - runManager->SetUserAction(new XXXPrimaryGeneratorAction);
  - runManager->SetUserAction(new XXXStackingAction);
  - runManager->SetUserAction(new XXXSteppingAction);
  - runManager->SetUserAction(new XXXEventAction);
  - runManager->SetUserAction(new XXXRunAction);

  - runManager->Initialize();

UserDetectorConstruction

- XXXDetectorConstruction.hh

```
class XXXDetectorConstruction : public G4VUserDetectorConstruction{
  public:
    XXXDetectorConstruction();
    virtual ~XXXDetectorConstruction();
  public:
    virtual G4VPhysicalVolume* Construct();
};
```

- XXXDetectorConstruction.cc: G4VPhysicalVolume* XXXDetectorConstruction::Construct()

```
G4Material* air = manager->FindOrBuildMaterial("G4_AIR");
G4Box* solidWorld = new G4Box("World", 5*m, 5*m, 5*m);
G4LogicalVolume* logicWorld = new G4LogicalVolume(solidWorld, air, "WorldLog");
G4VPhysicalVolume* physWorld = new G4PVPlacement(0,                 //no rotation
                                                 G4ThreeVector(),   //at (0,0,0)
                                                 logicWorld,        //its logical volume
                                                 "World",           //its name
                                                 0,                 //its mother  volume
                                                 false,             //no boolean operation
                                                 0,                 //copy number
                                                 checkOverlaps);
… new G4PVPlacement(…, logicBGO, "BGO", logicWorld, false, id, false);
…
logicWorld->SetVisAttributes (G4VisAttributes::Invisible);
logicBGO->SetVisAttributes (new G4VisAttributes(G4Colour(0/255., 0/255.,255/255.)));

XXXDetectorSD* sensitiveDetector = new XXXDetectorSD(…);
G4SDManager::GetSDMpointer()->AddNewDetector(sensitiveDetector);
logicBGO-> SetSensitiveDetector(sensitiveDetector);

return physWorld;
```

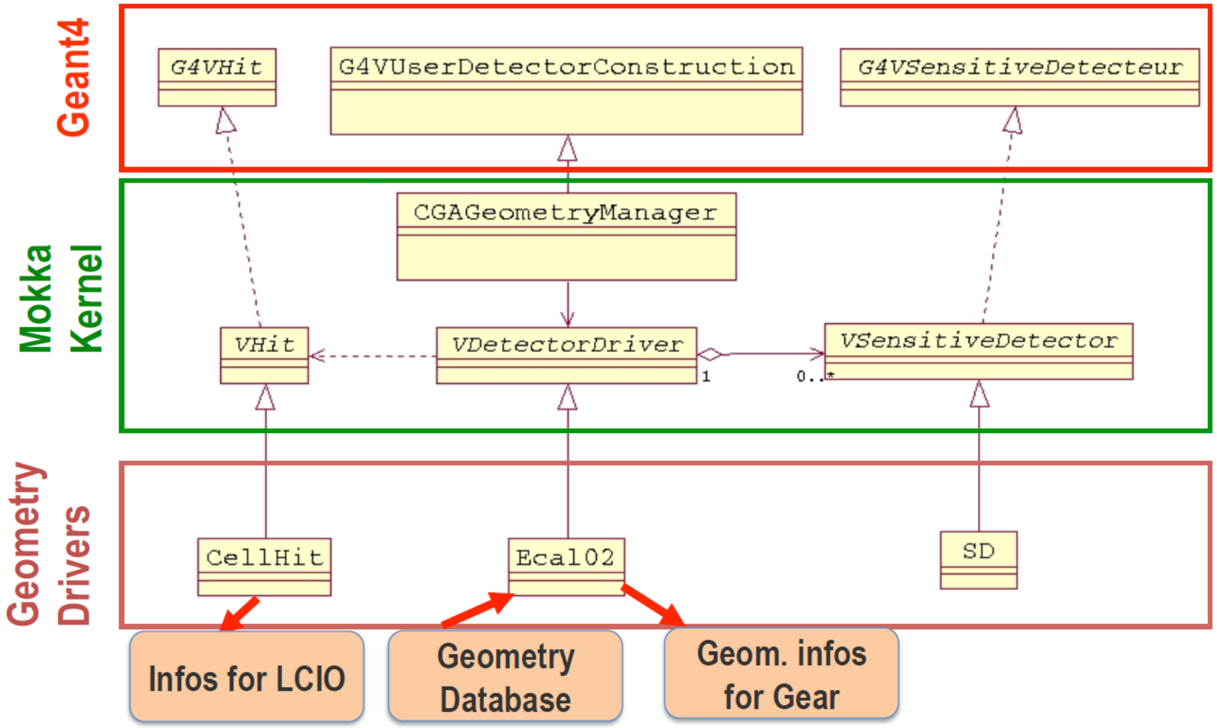
Material
Soild
Logical volume
Physical volume

Display

Sensitive detector

From MokkaC to DD4hep

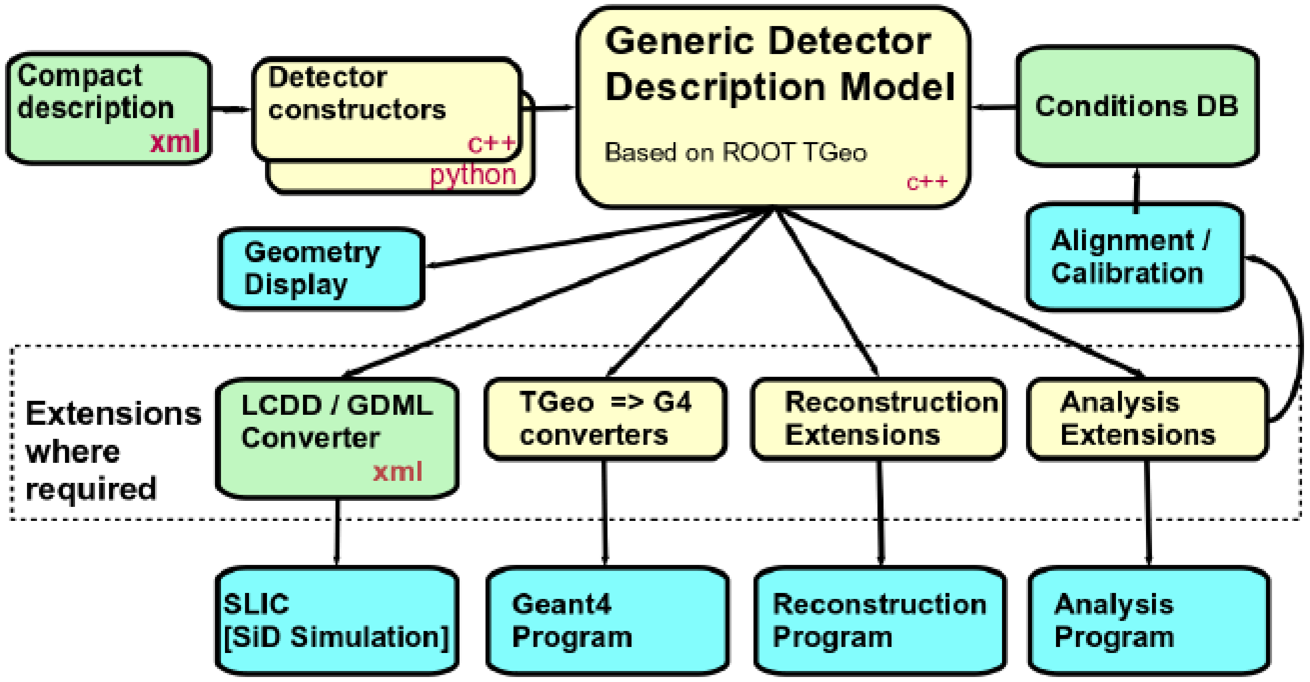
- Simulation toolkit
  - help users to build detectors in Geant4 and output results
- MokkaC—Mokka new improved version @CepC
  - Common interface between user and Geant4:
    - CGAGeometryManager: include storage
  - User define:
    - Driver, Sensitive detector, Hit



Emilia BECHEVA, LLR – Ecole Polytechnique, CNRS

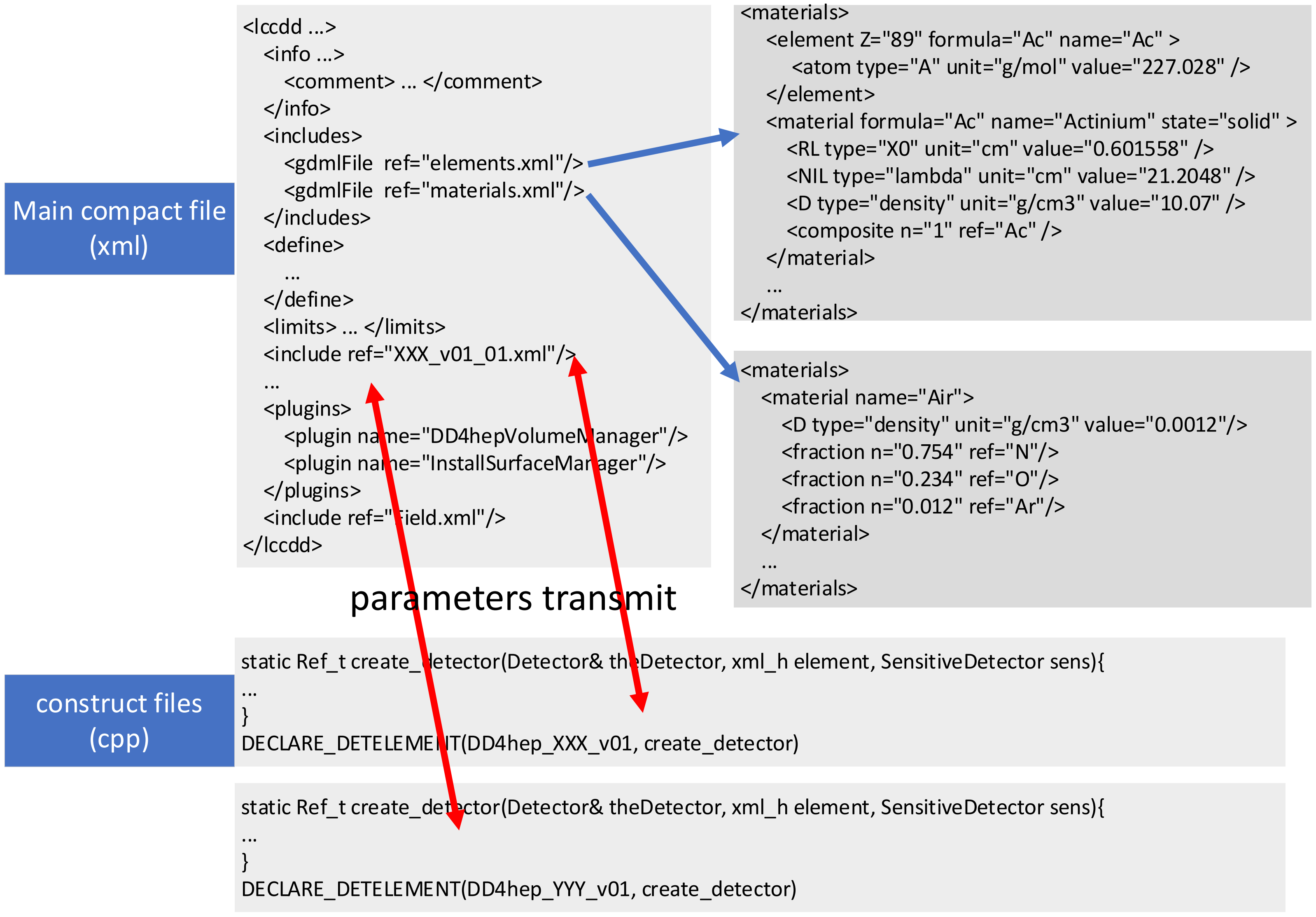
DD4hep

- Based on ROOT geometry classes
- More functions: easy to invoke
  - geometry (construction, display, overlap check, material scan etc.)
  - geometry converters
  - reconstruction extensions
  - analysis extensions



M. Frank, **DD4hep User Manual**

Build Detector by DD4hep



```
<lccdd ...>
   <info ...>
      <comment> ... </comment>
   </info>
   <includes>
      <gdmlFile  ref="elements.xml"/>
      <gdmlFile  ref="materials.xml"/>
   </includes>
   <define>
      ...
   </define>
   <limits> ... </limits>
   <include ref="XXX_v01_01.xml"/>
   ...
   <plugins>
      <plugin name="DD4hepVolumeManager"/>
      <plugin name="InstallSurfaceManager"/>
   </plugins>
   <include ref="Field.xml"/>
</lccdd>
```

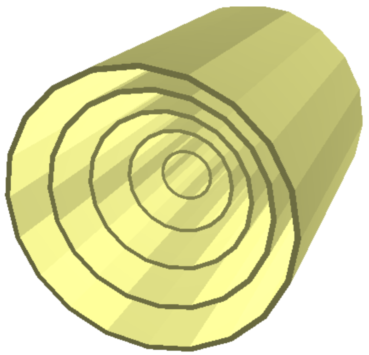
```
<materials>
    <element Z="89" formula="Ac" name="Ac" >
        <atom type="A" unit="g/mol" value="227.028" />
    </element>
    <material formula="Ac" name="Actinium" state="solid" >
       <RL type="X0" unit="cm" value="0.601558" />
       <NIL type="lambda" unit="cm" value="21.2048" />
       <D type="density" unit="g/cm3" value="10.07" />
       <composite n="1" ref="Ac" />
    </material>
    ...
</materials>
```

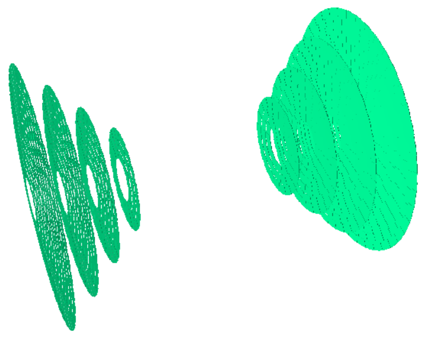
```
<materials>
   <material name="Air">
      <D type="density" unit="g/cm3" value="0.0012"/>
      <fraction n="0.754" ref="N"/>
      <fraction n="0.234" ref="O"/>
      <fraction n="0.012" ref="Ar"/>
   </material>
   ...
</materials>
```

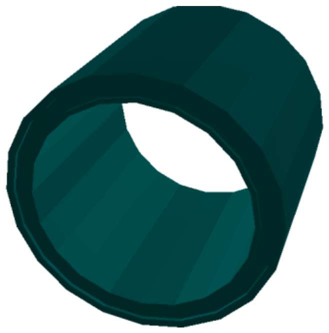
```
static Ref_t create_detector(Detector& theDetector, xml_h element, SensitiveDetector sens){
...
}
DECLARE_DETELEMENT(DD4hep_XXX_v01, create_detector)
```

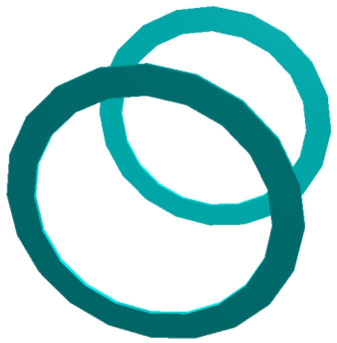
```
static Ref_t create_detector(Detector& theDetector, xml_h element, SensitiveDetector sens){
...
}
DECLARE_DETELEMENT(DD4hep_YYY_v01, create_detector)
```

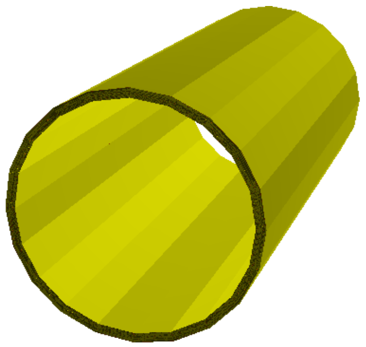
Basic Sub-detector in DD4hep

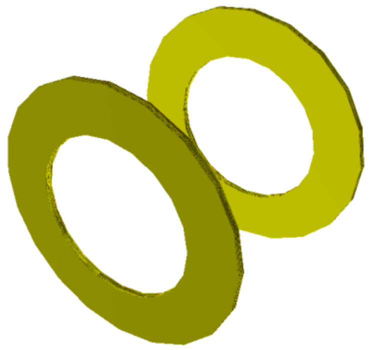

DD4hep_SiTrackerBarrel

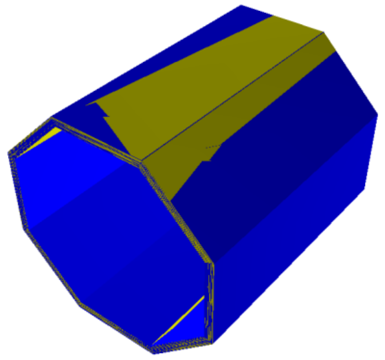

DD4hep_SiTrackerEndcap2

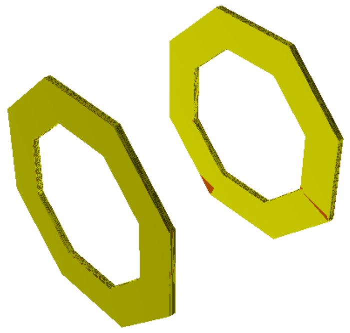

DD4hep_MultiLayerTracker

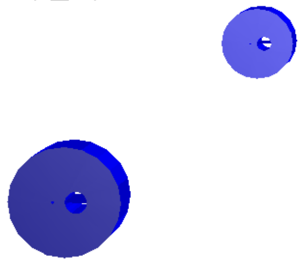

DD4hep_DiskTracker

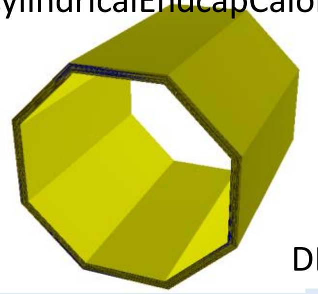

DD4hep_CylindricalBarrelCalorimeter


DD4hep_CylindricalEndcapCalorimeter


DD4hep_PolyhedraBarrelCalorimeter2


DD4hep_PolyhedraEndcapCalorimeter2


DD4hep_ForwardDetector

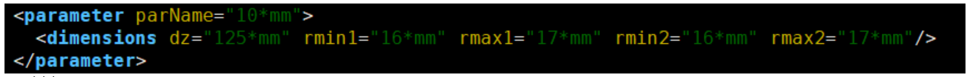

DD4hep_EcalBarrel

- lcgeo package also has built ILC sub-detector type, help us to move from MokkaC to DD4hep quickly.

DD4hep construction

static Ref_t create_detector(Detector& theDet, xml_h element, SensitiveDetector sens)

- Object prepare
  - DetElement det(const std::string& name, int id); DetElement(…)
  - Volume envelope = dd4hep::xml::createPlacedEnvelope(theDet, element, det);
  - …
- Parameters input



```
<parameter parName="10*mm">
  <dimensions dz="125*mm" rmin1="16*mm" rmax1="17*mm" rmin2="16*mm" rmax2="17*mm"/>
</parameter>
```

  - xml_det_t x_det = element;
  - xml_comp_t x(x_det.child(_Unicode(parameter)));
  - x.attr< double > (_Unicode(parName));
  - xml_comp_t x_dim(x_det.child(_U(dimensions)));
  - x_dim.dz(), x_dim.rmin1(), x_dim.rmax1(), x_dim.rmin2(), x_dim.rmax2()
- Volume placement
  - Box box(x_dim.dx(), x_dim.dy(), x_dim.dz()); … ⇔G4Box, …
  - Volume vol(x_det.nameStr()+"_Box", box, theDet.material(x_dim.materialStr()) ⇔G4LogicalVolume
  - PlacedVolume pv = envelope.placeVolume(vol, ⇔new G4PVPlacement
  - Transform3D(RotationZ(0.), Position(0, 0, 0))); ⇔G4Transform3D
  - pv.addPhysVolID("layer", layer_id).addPhysVolID("module", module_id)…;
  - vol.setVisAttributes(theDet, "TubeVis"); ⇔SetVisAttributes
  - DetElement subDE(det, "sub-component", x_det.id());
  - subDE.setPlacement(pv) ;
  - dd4hep::rec::VolPlane surf(vol, …) ;
  - dd4hep::rec::volSurfaceList(subDE) ->push_back(surf);
- Sensitive detector
  - sens.setType("SimpleCalorimeterSD");
  - vol.setSensitiveDetector(sens); ⇔SetSensitiveDetector
- Extension output
  - dd4hep::rec::ZPlanarData* zPlanarData = new dd4hep::rec::ZPlanarData ;
  - det.addExtension< ZPlanarData >(zPlanarData) ;

Solids

dd4hep	ROOT	Geant4
Box	TGeoBBox	G4Box
Tube	TGeoTube, TGeoTubeSeg	G4Tubs
CutTube	TGeoCtub	G4CutTubs
EllipticalTube	TGeoEltu	G4EllipticalTube
TwistedTube	TwistedTubeObject	G4TwistedTubs
Trd1, Trd2	TGeoTrd1, TGeoTrd2	G4Trd
Hyperboloid	TGeoHype	G4Hype
EightPointSolid	TGeoArb8, G4GenericTrap	G4GenericTrap
ExtrudedPolygon	TGeoXtru	G4ExtrudedSolid
PolyhedraRegular, Polyhedra	TGeoPgon	G4Polyhedra
Polycone	TGeoPcon	G4Polycone
Cone, ConeSegment	TGeoCone, TGeoConeSeg	G4Cons
Paraboloid	TGeoParaboloid	G4Paraboloid
Sphere	TGeoSphere	G4Sphere
Torus	TGeoTorus	G4Torus
Trap	TGeoTrap	G4Trap
TessellatedSolid	TGeoTessellated	G4TriangularFacet G4QuadrangularFacet
	TGeoScaledShape	G4ReflectedSolid
PseudoTrap TruncatedTube SubtractionSolid UnionSolid IntersectionSolid	TGeoCompositeShape	G4Ellipsoid G4SubtractionSolid G4UnionSolid G4IntersectionSolid

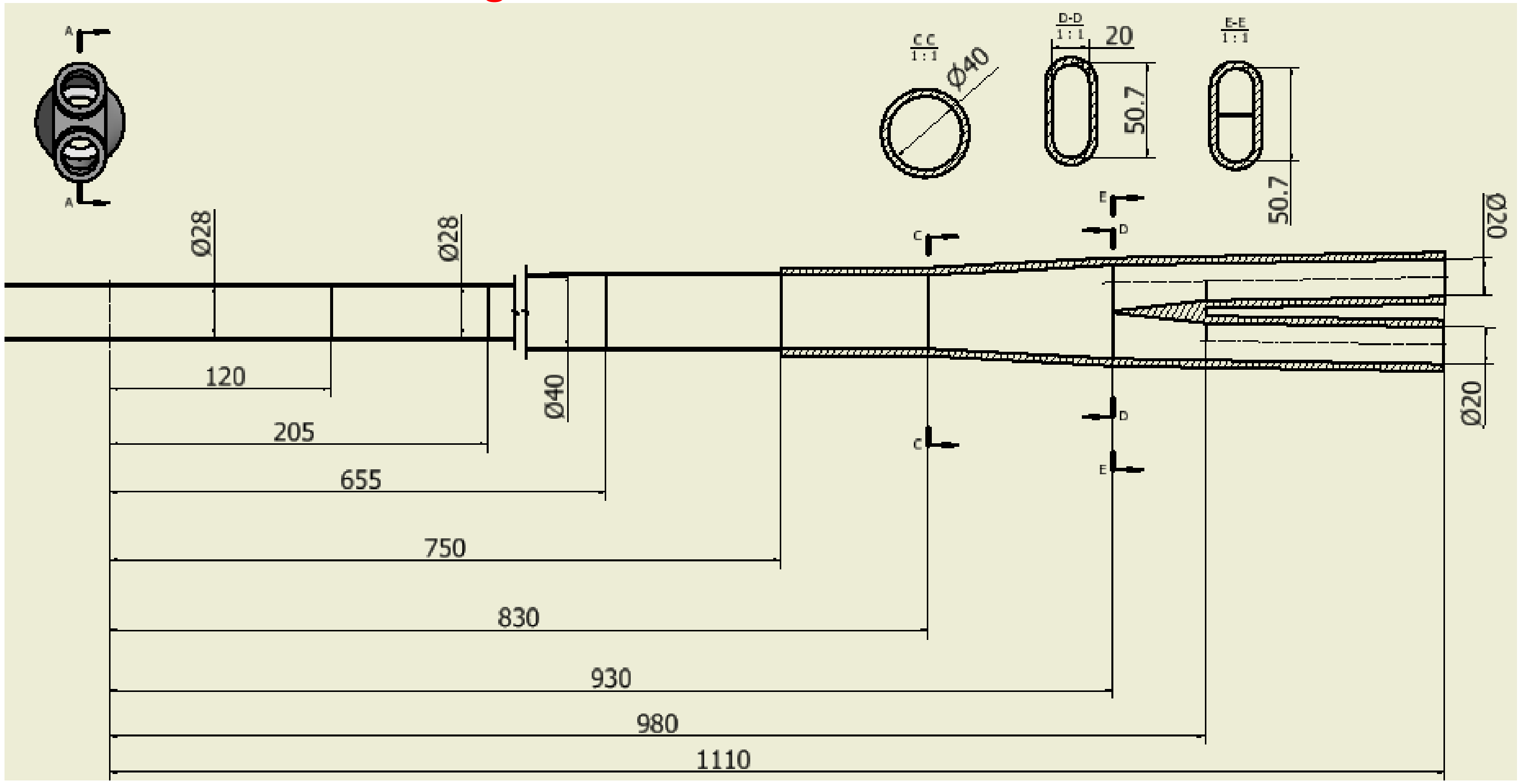
- For sepecial solid XXX, beside class XXX, class TGeoXXX and class G4XXX, convertShape<TGeoXXX>(shape) also needed in Geant4Converter::handleSolid(…), not recommended

An Example

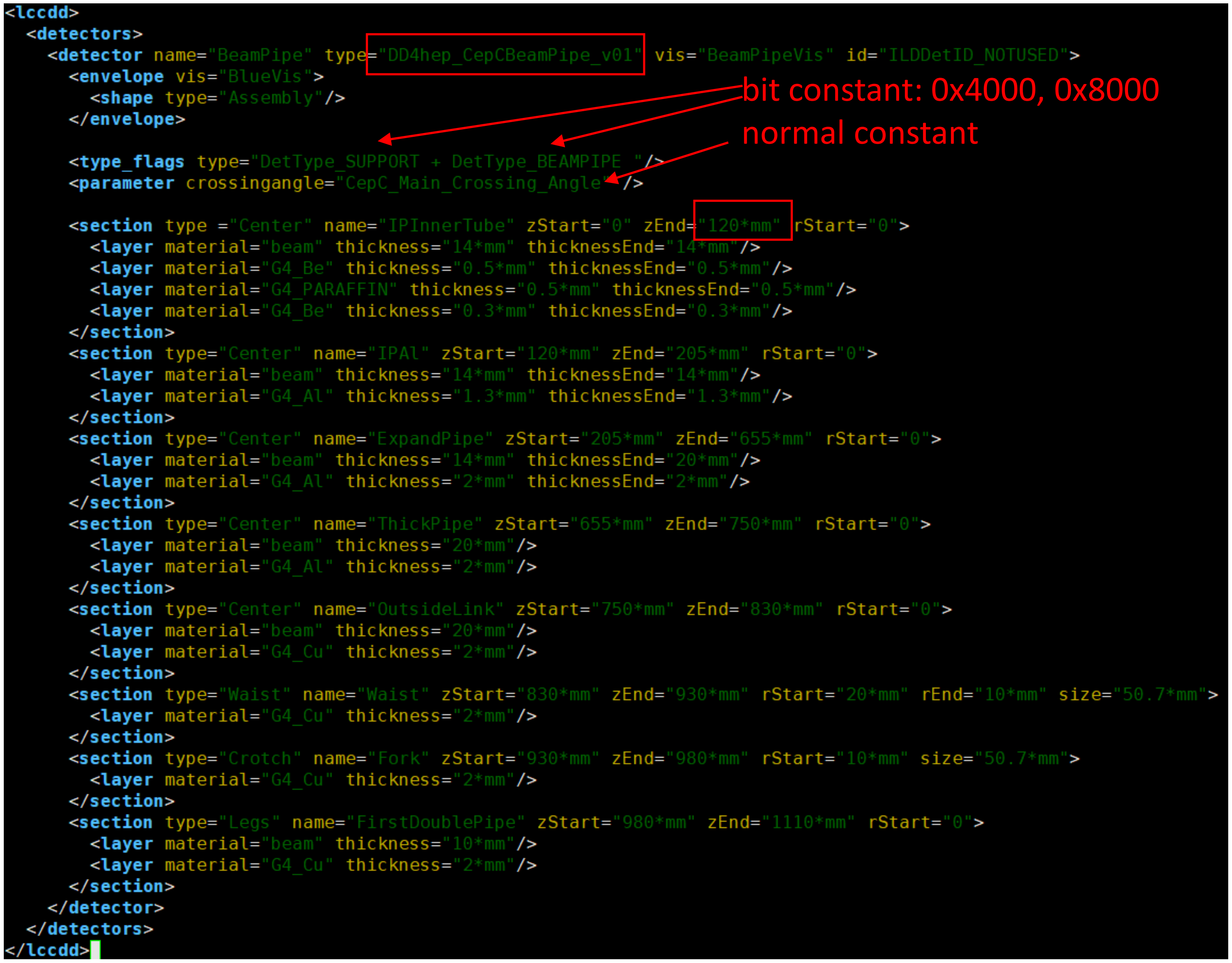
- Mokka::Tube → Mokka::CepCBeamPipe → DD4hep::CepCBeamPipe_v01

  database **xml(user manage)** **xml(dd4hep manage)**
- MokkaC driver: http://cepcgit.ihep.ac.cn/cepcsoft/MokkaC/-/blob/master/source/Geometry/CEPC/src/CepCBeamPipe.cc
- DD4hep createment: https://github.com/fucd/CEPCSW/blob/master/Detector/Reference/src/other/CepCBeamPipe_v01_geo.cpp

one old design from BAI Sha

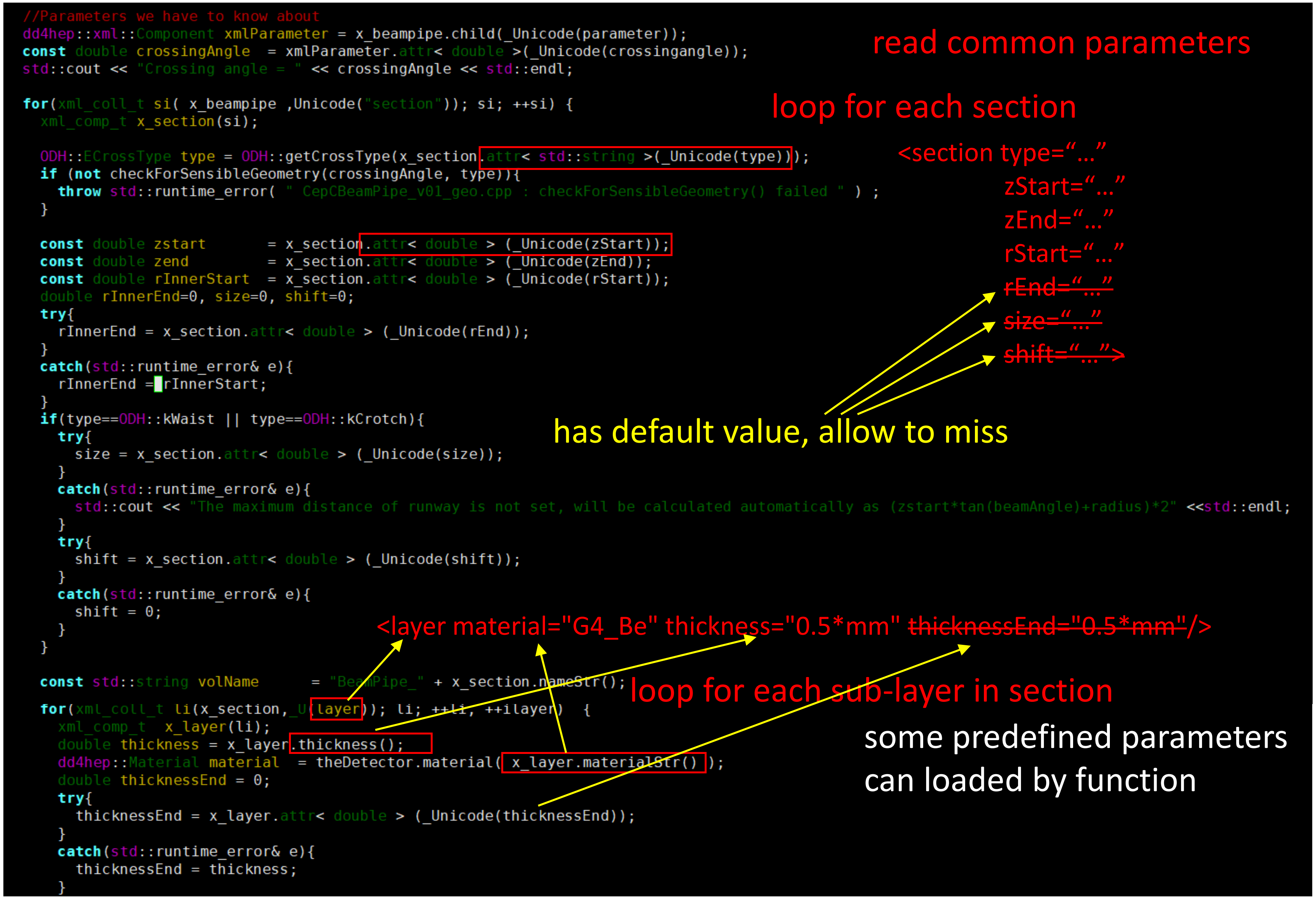


Compact



```xml
<lccdd>
  <detectors>
    <detector name="BeamPipe" type="DD4hep_CepCBeamPipe_v01" vis="BeamPipeVis" id="ILDDetID_NOTUSED">
      <envelope vis="BlueVis">
        <shape type="Assembly"/>
      </envelope>

      <type_flags type="DetType_SUPPORT + DetType_BEAMPIPE "/>
      <parameter crossingangle="CepC_Main_Crossing_Angle" />

      <section type ="Center" name="IPInnerTube" zStart="0" zEnd="120*mm" rStart="0">
        <layer material="beam" thickness="14*mm" thicknessEnd="14*mm"/>
        <layer material="G4_Be" thickness="0.5*mm" thicknessEnd="0.5*mm"/>
        <layer material="G4_PARAFFIN" thickness="0.5*mm" thicknessEnd="0.5*mm"/>
        <layer material="G4_Be" thickness="0.3*mm" thicknessEnd="0.3*mm"/>
      </section>
      <section type="Center" name="IPAl" zStart="120*mm" zEnd="205*mm" rStart="0">
        <layer material="beam" thickness="14*mm" thicknessEnd="14*mm"/>
        <layer material="G4_Al" thickness="1.3*mm" thicknessEnd="1.3*mm"/>
      </section>
      <section type="Center" name="ExpandPipe" zStart="205*mm" zEnd="655*mm" rStart="0">
        <layer material="beam" thickness="14*mm" thicknessEnd="20*mm"/>
        <layer material="G4_Al" thickness="2*mm" thicknessEnd="2*mm"/>
      </section>
      <section type="Center" name="ThickPipe" zStart="655*mm" zEnd="750*mm" rStart="0">
        <layer material="beam" thickness="20*mm"/>
        <layer material="G4_Al" thickness="2*mm"/>
      </section>
      <section type="Center" name="OutsideLink" zStart="750*mm" zEnd="830*mm" rStart="0">
        <layer material="beam" thickness="20*mm"/>
        <layer material="G4_Cu" thickness="2*mm"/>
      </section>
      <section type="Waist" name="Waist" zStart="830*mm" zEnd="930*mm" rStart="20*mm" rEnd="10*mm" size="50.7*mm">
        <layer material="G4_Cu" thickness="2*mm"/>
      </section>
      <section type="Crotch" name="Fork" zStart="930*mm" zEnd="980*mm" rStart="10*mm" size="50.7*mm">
        <layer material="G4_Cu" thickness="2*mm"/>
      </section>
      <section type="Legs" name="FirstDoublePipe" zStart="980*mm" zEnd="1110*mm" rStart="0">
        <layer material="beam" thickness="10*mm"/>
        <layer material="G4_Cu" thickness="2*mm"/>
      </section>
    </detector>
  </detectors>
</lccdd>
```

Parameter Parser



```
//Parameters we have to know about
dd4hep::xml::Component xmlParameter = x_beampipe.child(_Unicode(parameter));
const double crossingAngle  = xmlParameter.attr< double >(_Unicode(crossingangle));
std::cout << "Crossing angle = " << crossingAngle << std::endl;

for(xml_coll_t si( x_beampipe ,Unicode("section")); si; ++si) {
  xml_comp_t x_section(si);

  ODH::ECrossType type = ODH::getCrossType(x_section.attr< std::string >(_Unicode(type)));
  if (not checkForSensibleGeometry(crossingAngle, type)){
    throw std::runtime_error( " CepCBeamPipe_v01_geo.cpp : checkForSensibleGeometry() failed " ) ;
  }

  const double zstart       = x_section.attr< double > (_Unicode(zStart));
  const double zend         = x_section.attr< double > (_Unicode(zEnd));
  const double rInnerStart  = x_section.attr< double > (_Unicode(rStart));
  double rInnerEnd=0, size=0, shift=0;
  try{
    rInnerEnd = x_section.attr< double > (_Unicode(rEnd));
  }
  catch(std::runtime_error& e){
    rInnerEnd = rInnerStart;
  }
  if(type==ODH::kWaist || type==ODH::kCrotch){
    try{
      size = x_section.attr< double > (_Unicode(size));
    }
    catch(std::runtime_error& e){
      std::cout << "The maximum distance of runway is not set, will be calculated automatically as (zstart*tan(beamAngle)+radius)*2" <<std::endl;
    }
    try{
      shift = x_section.attr< double > (_Unicode(shift));
    }
    catch(std::runtime_error& e){
      shift = 0;
    }
  }

  const std::string volName      = "BeamPipe_" + x_section.nameStr();

  for(xml_coll_t li(x_section,_U(layer)); li; ++li, ++ilayer)  {
    xml_comp_t  x_layer(li);
    double thickness = x_layer.thickness();
    dd4hep::Material material    = theDetector.material( x_layer.materialStr() );
    double thicknessEnd = 0;
    try{
      thicknessEnd = x_layer.attr< double > (_Unicode(thicknessEnd));
    }
    catch(std::runtime_error& e){
      thicknessEnd = thickness;
    }
```

read common parameters

loop for each section

<section type="…"
zStart="…"
zEnd="…"
rStart="…"
~~rEnd="…"~~
~~size="…"~~
~~shift="…"~~>

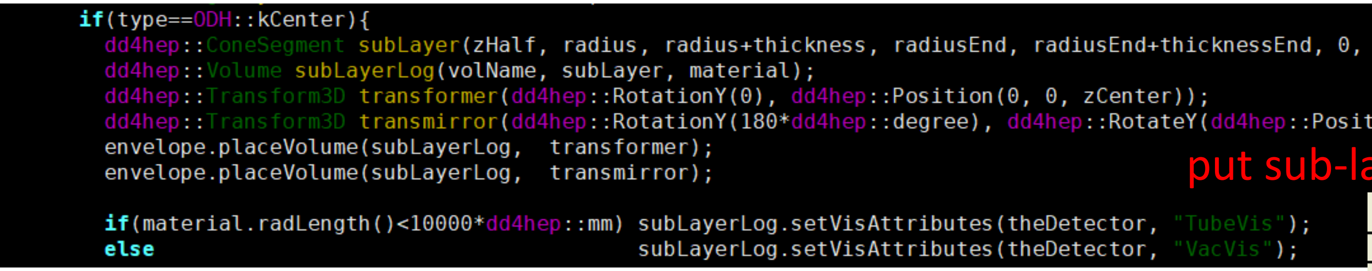
has default value, allow to miss

<layer material="G4_Be" thickness="0.5*mm" ~~thicknessEnd="0.5*mm"~~/>

loop for each sub-layer in section

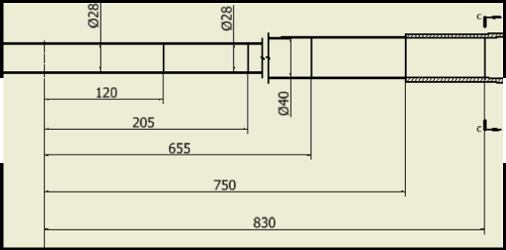
some predefined parameters can loaded by function

Volumes



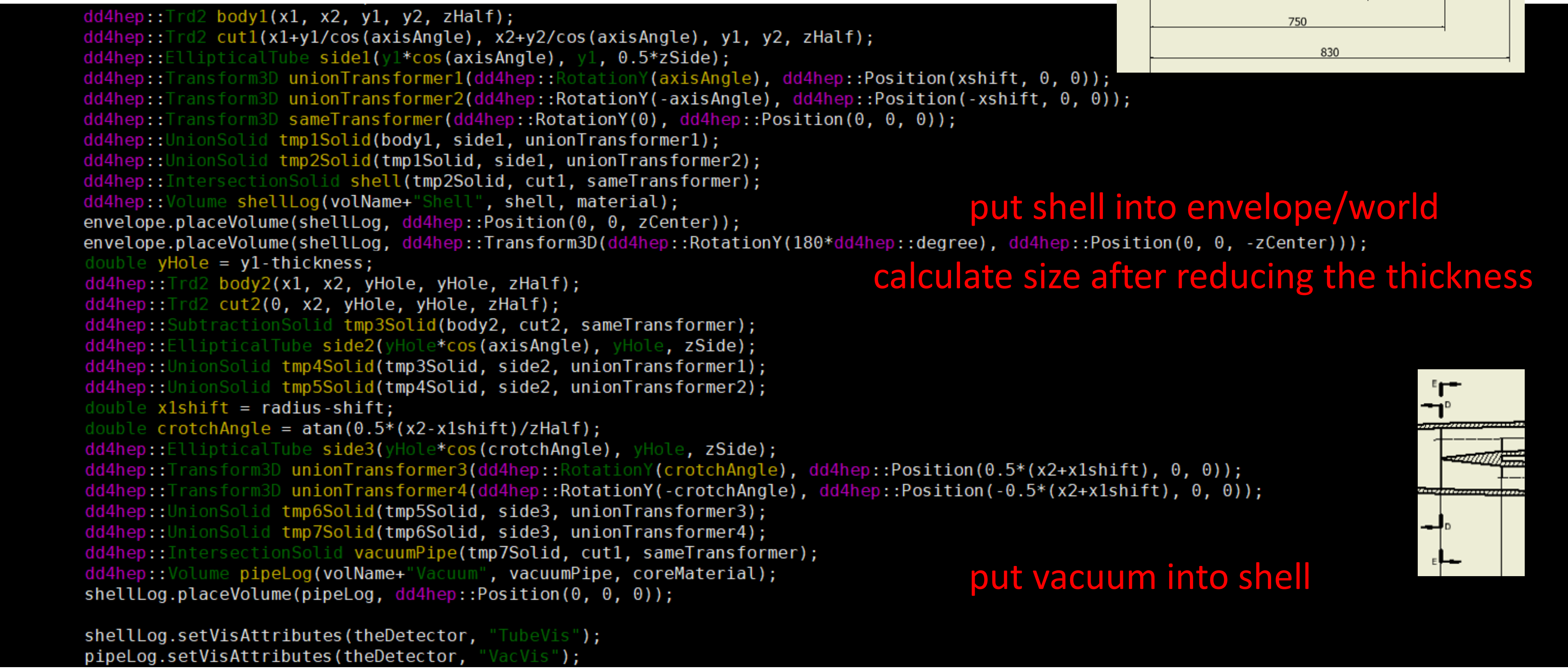
```
  if(type==ODH::kCenter){
    dd4hep::ConeSegment subLayer(zHalf, radius, radius+thickness, radiusEnd, radiusEnd+thicknessEnd, 0, 360*dd4hep::degree);
    dd4hep::Volume subLayerLog(volName, subLayer, material);
    dd4hep::Transform3D transformer(dd4hep::RotationY(0), dd4hep::Position(0, 0, zCenter));
    dd4hep::Transform3D transmirror(dd4hep::RotationY(180*dd4hep::degree), dd4hep::RotateY(dd4hep::Position(0, 0, zCenter), 180*dd4hep::degree));
    envelope.placeVolume(subLayerLog,  transformer);
    envelope.placeVolume(subLayerLog,  transmirror);

    if(material.radLength()<10000*dd4hep::mm) subLayerLog.setVisAttributes(theDetector, "TubeVis");
    else                                      subLayerLog.setVisAttributes(theDetector, "VacVis");
```

put sub-layer into envelope/world



type==ODH::kCrotch

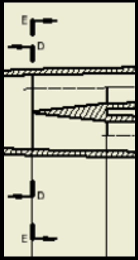
…



```
    dd4hep::Trd2 body1(x1, x2, y1, y2, zHalf);
    dd4hep::Trd2 cut1(x1+y1/cos(axisAngle), x2+y2/cos(axisAngle), y1, y2, zHalf);
    dd4hep::EllipticalTube side1(y1*cos(axisAngle), y1, 0.5*zSide);
    dd4hep::Transform3D unionTransformer1(dd4hep::RotationY(axisAngle), dd4hep::Position(xshift, 0, 0));
    dd4hep::Transform3D unionTransformer2(dd4hep::RotationY(-axisAngle), dd4hep::Position(-xshift, 0, 0));
    dd4hep::Transform3D sameTransformer(dd4hep::RotationY(0), dd4hep::Position(0, 0, 0));
    dd4hep::UnionSolid tmp1Solid(body1, side1, unionTransformer1);
    dd4hep::UnionSolid tmp2Solid(tmp1Solid, side1, unionTransformer2);
    dd4hep::IntersectionSolid shell(tmp2Solid, cut1, sameTransformer);
    dd4hep::Volume shellLog(volName+"Shell", shell, material);
    envelope.placeVolume(shellLog, dd4hep::Position(0, 0, zCenter));
    envelope.placeVolume(shellLog, dd4hep::Transform3D(dd4hep::RotationY(180*dd4hep::degree), dd4hep::Position(0, 0, -zCenter)));
    double yHole = y1-thickness;
    dd4hep::Trd2 body2(x1, x2, yHole, yHole, zHalf);
    dd4hep::Trd2 cut2(0, x2, yHole, yHole, zHalf);
    dd4hep::SubtractionSolid tmp3Solid(body2, cut2, sameTransformer);
    dd4hep::EllipticalTube side2(yHole*cos(axisAngle), yHole, zSide);
    dd4hep::UnionSolid tmp4Solid(tmp3Solid, side2, unionTransformer1);
    dd4hep::UnionSolid tmp5Solid(tmp4Solid, side2, unionTransformer2);
    double x1shift = radius-shift;
    double crotchAngle = atan(0.5*(x2-x1shift)/zHalf);
    dd4hep::EllipticalTube side3(yHole*cos(crotchAngle), yHole, zSide);
    dd4hep::Transform3D unionTransformer3(dd4hep::RotationY(crotchAngle), dd4hep::Position(0.5*(x2+x1shift), 0, 0));
    dd4hep::Transform3D unionTransformer4(dd4hep::RotationY(-crotchAngle), dd4hep::Position(-0.5*(x2+x1shift), 0, 0));
    dd4hep::UnionSolid tmp6Solid(tmp5Solid, side3, unionTransformer3);
    dd4hep::UnionSolid tmp7Solid(tmp6Solid, side3, unionTransformer4);
    dd4hep::IntersectionSolid vacuumPipe(tmp7Solid, cut1, sameTransformer);
    dd4hep::Volume pipeLog(volName+"Vacuum", vacuumPipe, coreMaterial);
    shellLog.placeVolume(pipeLog, dd4hep::Position(0, 0, 0));

    shellLog.setVisAttributes(theDetector, "TubeVis");
    pipeLog.setVisAttributes(theDetector, "VacVis");
```

put shell into envelope/world

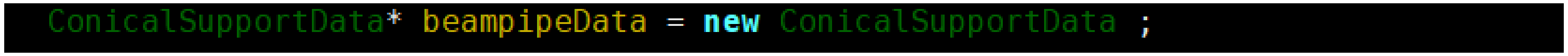
calculate size after reducing the thickness



put vacuum into shell

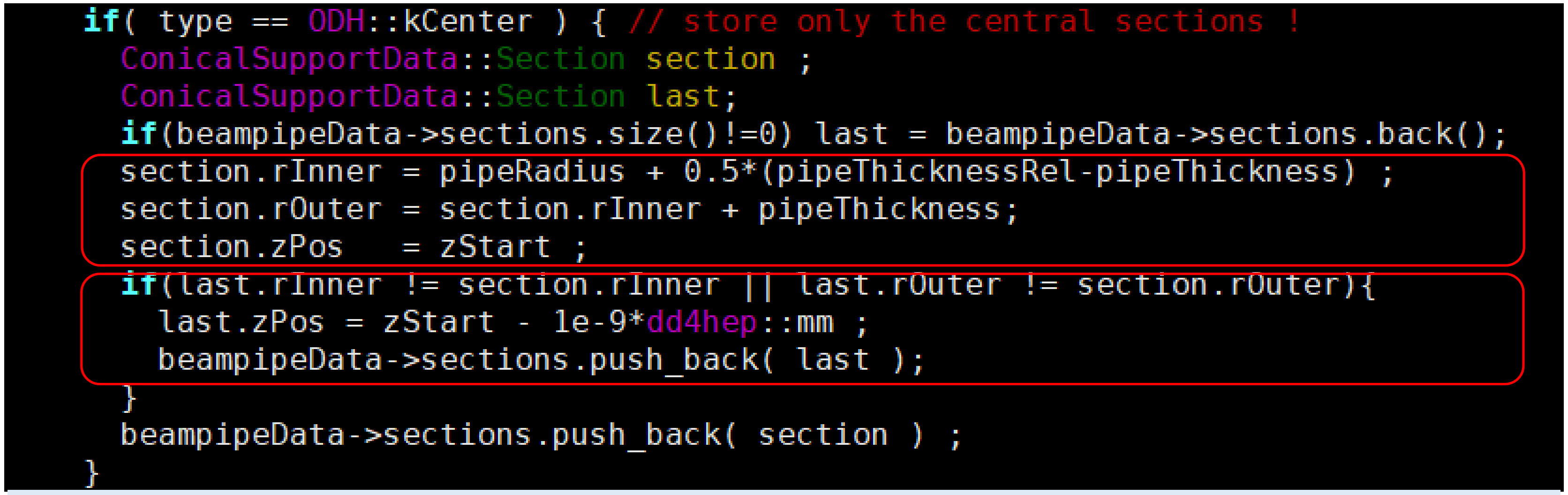
DetectorData

- dd4hep::recConicalSupportData



```
ConicalSupportData* beampipeData = new ConicalSupportData ;
```

- loop section

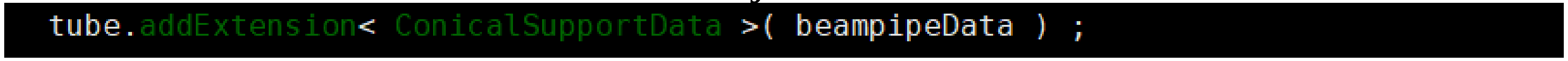


```
if( type == ODH::kCenter ) { // store only the central sections !
  ConicalSupportData::Section section ;
  ConicalSupportData::Section last;
  if(beampipeData->sections.size()!=0) last = beampipeData->sections.back();
  section.rInner = pipeRadius + 0.5*(pipeThicknessRel-pipeThickness) ;
  section.rOuter = section.rInner + pipeThickness;
  section.zPos   = zStart ;
  if(last.rInner != section.rInner || last.rOuter != section.rOuter){
    last.zPos = zStart - 1e-9*dd4hep::mm ;
    beampipeData->sections.push_back( last );
  }
  beampipeData->sections.push_back( section ) ;
}
```

⇒temporary: use effective thickness and keep center same, since only one fixed beryllium layer in current reconstruction

⇒if discontinuous, add one very short pipe as passing

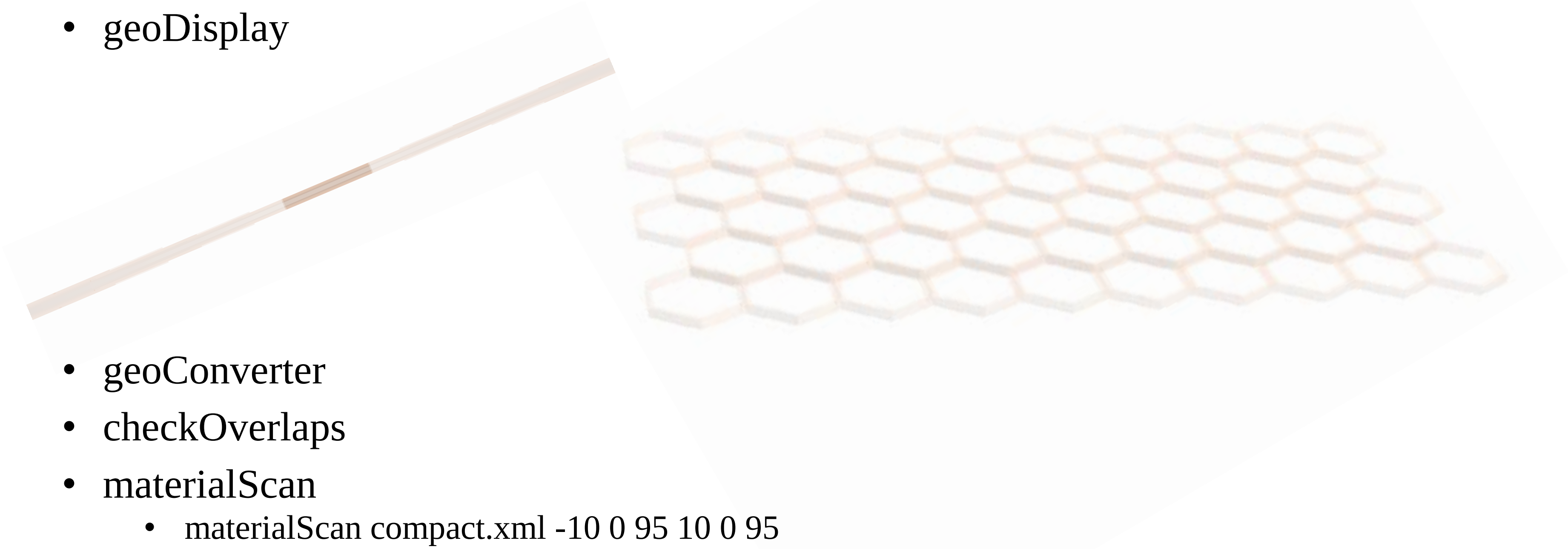
- add extension into DetElement object



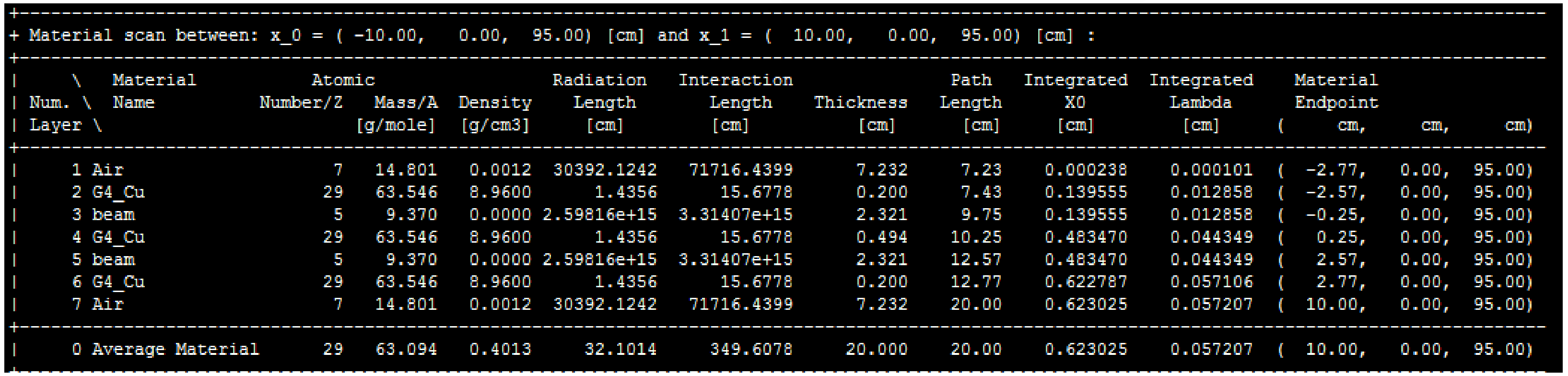
```
tube.addExtension< ConicalSupportData >( beampipeData ) ;
```

Check Tools

- export LD_LIBRARY_PATH=$CEPCSW/InstallArea/lib:$ LD_LIBRARY_PATH
- geoDisplay



- geoConverter
- checkOverlaps
- materialScan
  - materialScan compact.xml -10 0 95 10 0 95

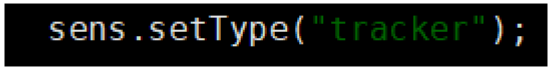


```
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
+ Material scan between: x_0 = ( -10.00,   0.00,  95.00) [cm] and x_1 = (  10.00,   0.00,  95.00) [cm] :
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
|     \   Material          Atomic                Radiation   Interaction              Path   Integrated  Integrated    Material
| Num. \  Name          Number/Z   Mass/A  Density    Length       Length    Thickness   Length      X0        Lambda      Endpoint
| Layer \                         [g/mole]  [g/cm3]     [cm]         [cm]         [cm]     [cm]      [cm]       [cm]     (      cm,      cm,      cm)
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
|     1 Air                   7   14.801   0.0012  30392.1242  71716.4399      7.232     7.23   0.000238    0.000101  (  -2.77,    0.00,   95.00)
|     2 G4_Cu                29   63.546   8.9600      1.4356     15.6778      0.200     7.43   0.139555    0.012858  (  -2.57,    0.00,   95.00)
|     3 beam                  5    9.370   0.0000 2.59816e+15 3.31407e+15      2.321     9.75   0.139555    0.012858  (  -0.25,    0.00,   95.00)
|     4 G4_Cu                29   63.546   8.9600      1.4356     15.6778      0.494    10.25   0.483470    0.044349  (   0.25,    0.00,   95.00)
|     5 beam                  5    9.370   0.0000 2.59816e+15 3.31407e+15      2.321    12.57   0.483470    0.044349  (   2.57,    0.00,   95.00)
|     6 G4_Cu                29   63.546   8.9600      1.4356     15.6778      0.200    12.77   0.622787    0.057106  (   2.77,    0.00,   95.00)
|     7 Air                   7   14.801   0.0012  30392.1242  71716.4399      7.232    20.00   0.623025    0.057207  (  10.00,    0.00,   95.00)
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
|     0 Average Material     29   63.094   0.4013     32.1014    349.6078     20.000    20.00   0.623025    0.057207  (  10.00,    0.00,   95.00)
+--------------------------------------------------------------------------------------------------------------------------------------------------------------
```

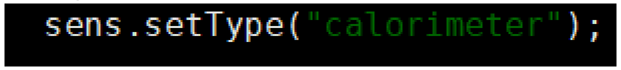
- etc.

Sensitive Detector

- In the XML description
  - <detector name="TPC" type="TPC10" vis="TPCVis" id="ILDDetID_TPC" limits="Tracker_limits" readout="TPCCollection" insideTrackingVolume="true">
  -     <sensitive type="tracker"/>   VS
  - …
  - </detector>
  - <readouts>
  -     <readout name="TPCCollection">
  -         <id>system:5,side:2,layer:9,module:8,sensor:8</id>
  -     </readout>
  - </readouts>
  - <detector name="EcalBarrel" type="SEcal05_Barrel" id="ILDDetID_ECAL" readout="EcalBarrelCollection" vis="BlueVis" >
  -     …
  -         <slice material = "Si" thickness = "Ecal_Si_thickness" sensitive = "yes" vis="Invisible"/>
  -         …
  - </detector>
  - <readouts>
  -     <readout name="EcalBarrelCollection">
  -         <segmentation type="MegatileLayerGridXY"/>
  -         <id>system:5,module:3,stave:4,tower:5,layer:6,wafer:6,cellX:32:-16,cellY:-16</id>
  -     </readout>
  - </readouts>
- Notice: Volume::setSensitiveDetector(SensitiveDetector sens) still needed in construction



```
sens.setType("tracker");
```

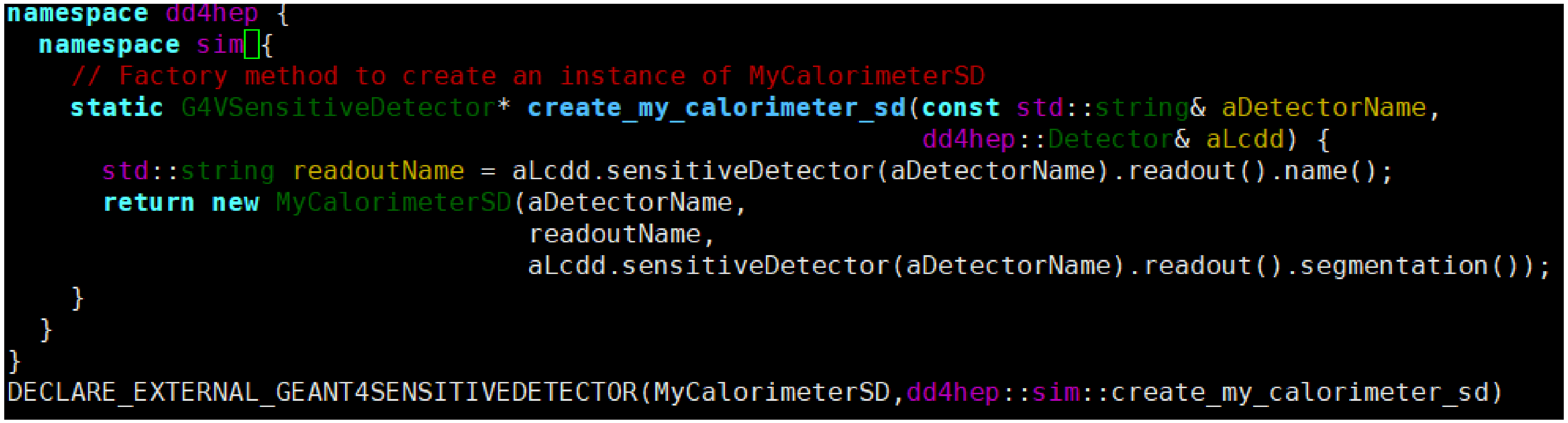


```
sens.setType("calorimeter");
```

User-define

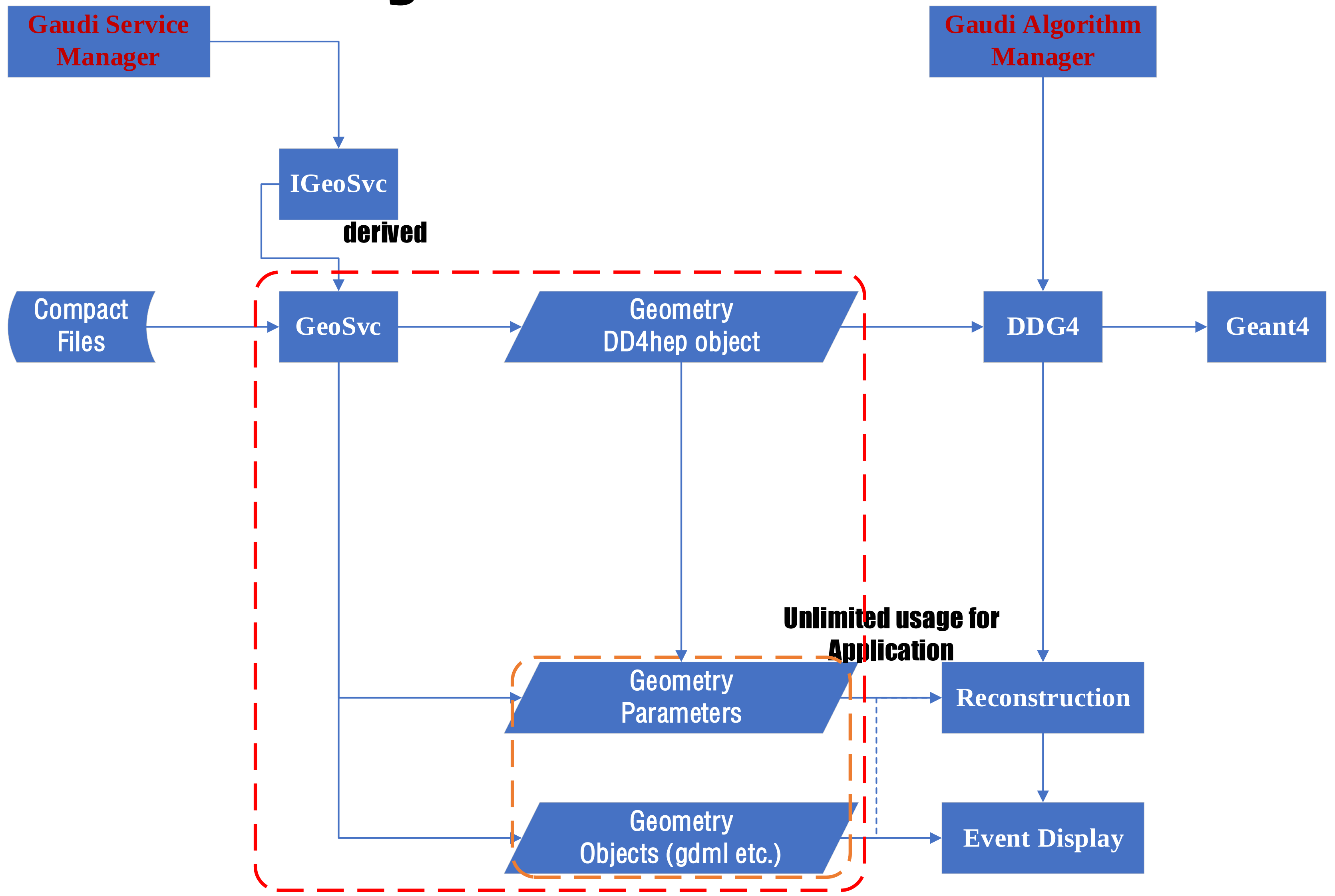
- class MyCalorimeterSD : public G4VSensitiveDetector {
-   public:
-     MyCalorimeterSD(std::string aDetName, std::string aReadoutName, dd4hep::Segmentation aSeg);
-     ~MyCalorimeterSD();
-     virtual void Initialize(4HCofThisEvent* aHitsCollections) final;
-     virtual bool ProcessHits(G4Step* aStep, G4TouchableHistory*) final;
- private:
-     G4THitsCollection<MyCaloHit>* calorimeterCollection;
- }

similar with standalone Geant4 simulation



```
namespace dd4hep {
  namespace sim {
    // Factory method to create an instance of MyCalorimeterSD
    static G4VSensitiveDetector* create_my_calorimeter_sd(const std::string& aDetectorName,
                                                          dd4hep::Detector& aLcdd) {
      std::string readoutName = aLcdd.sensitiveDetector(aDetectorName).readout().name();
      return new MyCalorimeterSD(aDetectorName,
                                 readoutName,
                                 aLcdd.sensitiveDetector(aDetectorName).readout().segmentation());
    }
  }
}
DECLARE_EXTERNAL_GEANT4SENSITIVEDETECTOR(MyCalorimeterSD,dd4hep::sim::create_my_calorimeter_sd)
```

Geometry Transfer



- Extension is used as geometry parameters carrier
- In future, dd4hep::rec::volSurfaceList is possible to help reconstruction to extraplate

Summary

- To describe a sub-detector
  - compact file (xml) to input geometry parameters
  - construction include: parameter parsing, volume placement, visualization attribution, sensitive detector, extension data, surface definition
- At first step
  - design compact file and parser
  - Use basic solids and basic sensitive detector
  - Ignore extension data and surface definition
  - Visualize check and show
- Face to future reconstruction
  - extension data
  - surface definition
- Meet Issues
  - Contact to CEPCSW groupers or discuss at github

- **Happy to use CEPCSW!**
- **Thanks very much for attention!**