TPC and STT geometry using DD4HEP in Aurora

Vijayanand KV
Scientific advisor: Andrey Sokolov

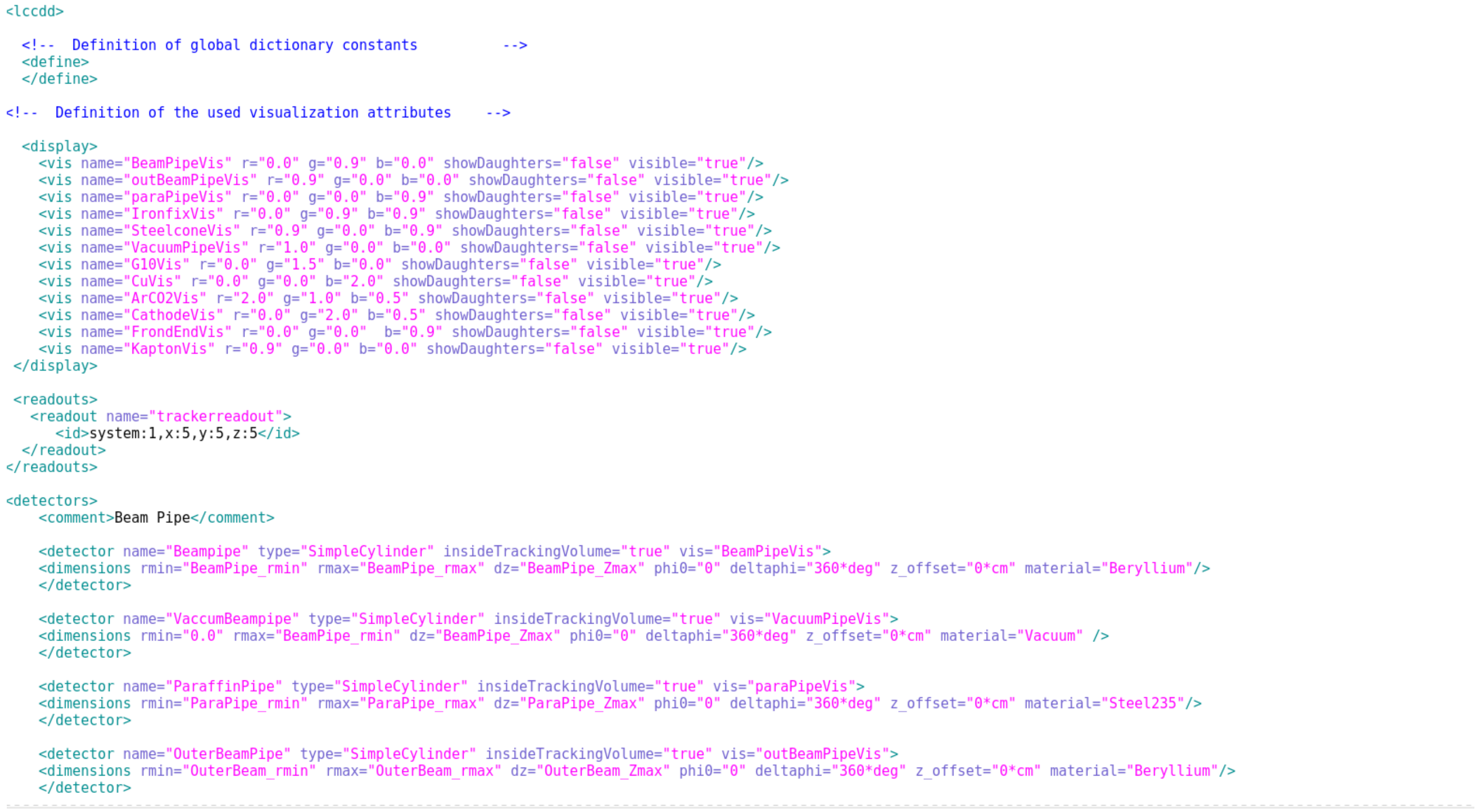
04-10-2018

Introduction

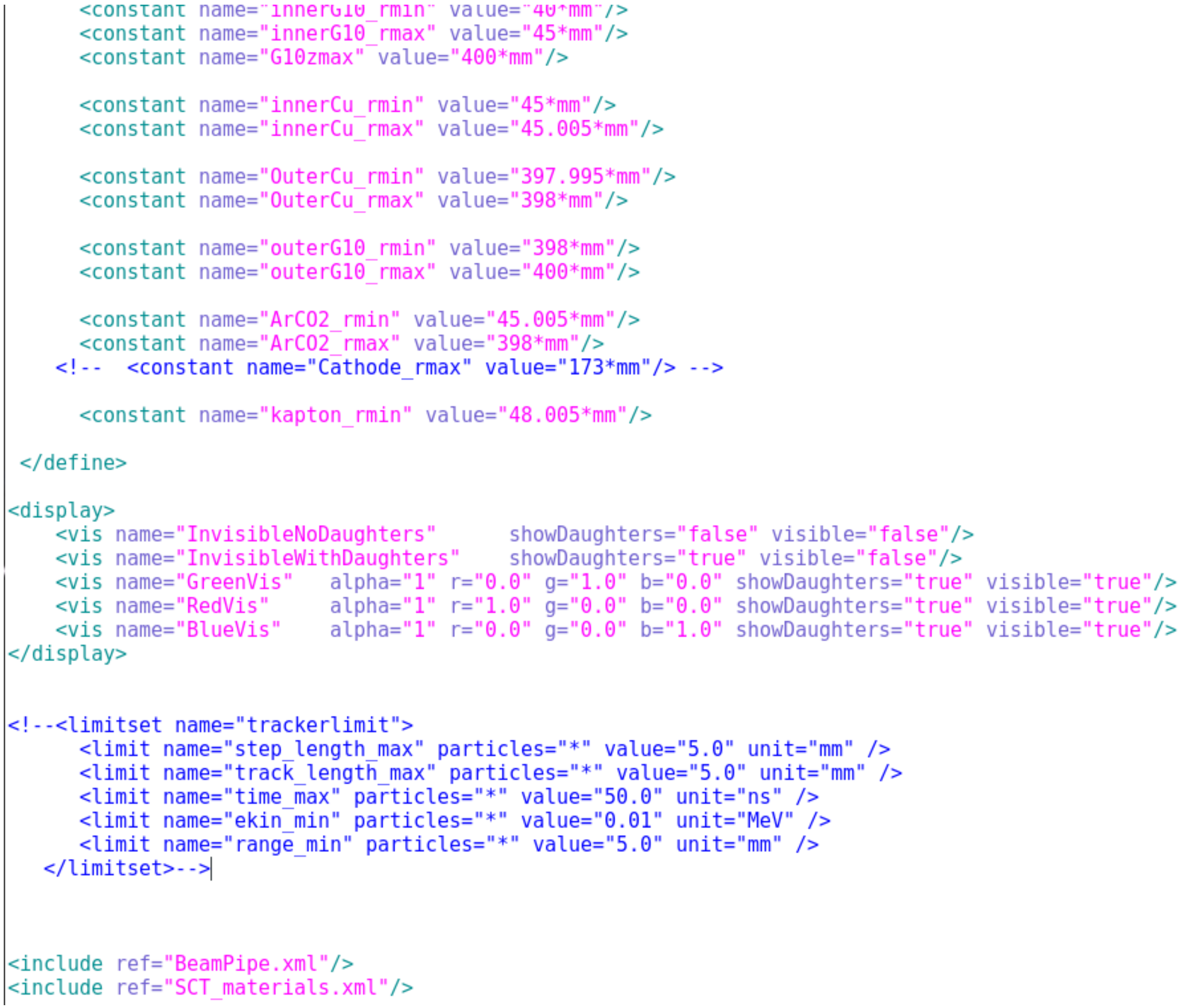
- DD4hep is a user friendly software for detector geometry construction.
- main components are the ROOT geometry package, which is used for construction and visualization of geometry, and the Geant4 simulation toolkit, which can be interfaced via DD4hep to perform detector simulation in complex detector designs.
- DD4hep have two type of scripts. One in xml for the description of geometry parameter and another one in c++ for user defined geometry.

Geometry of TPC in DD4hep.

- The script for geometry of TPC in DD4hep is very simple.
- There is no need of user defined geometry in TPC

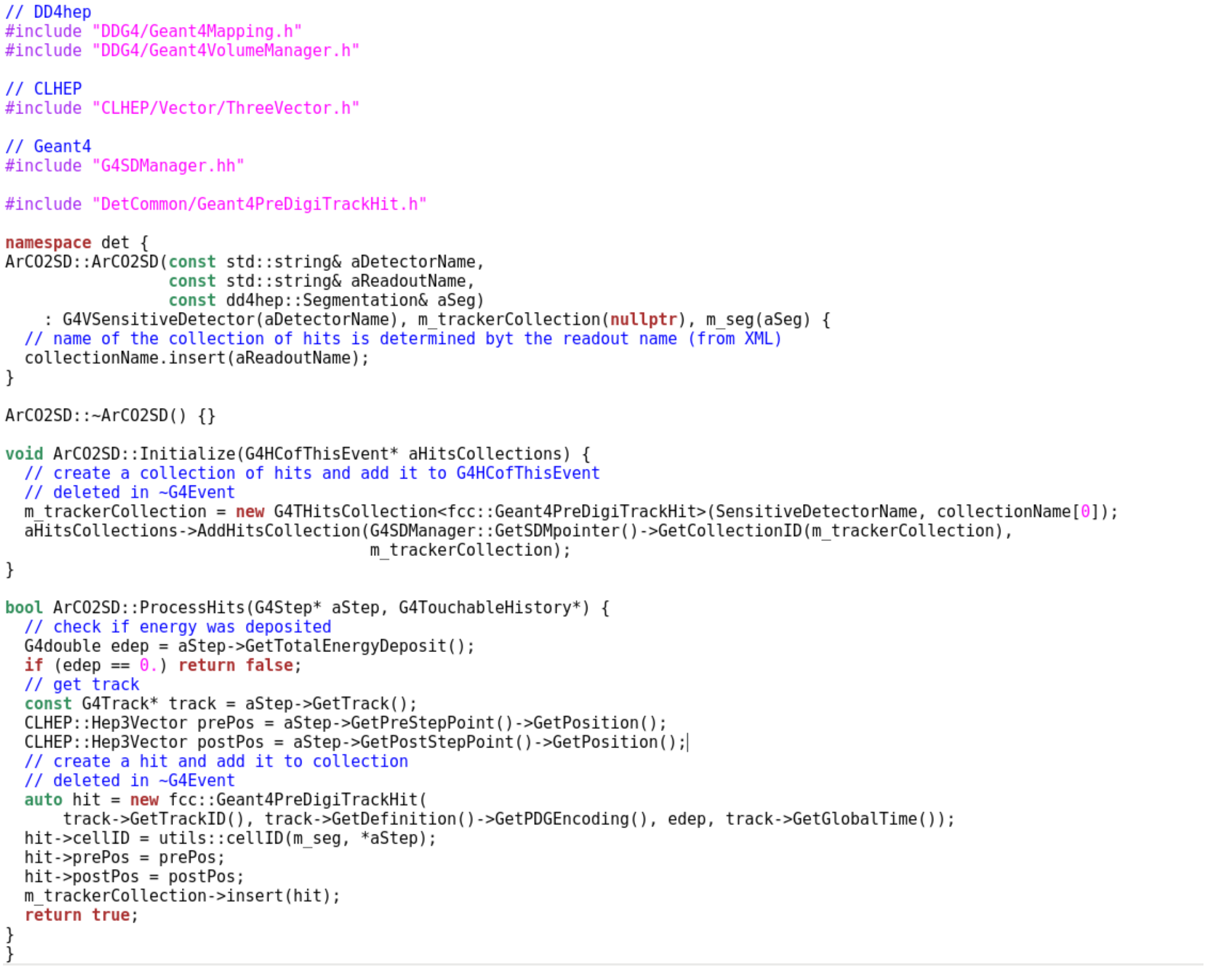


```
<lccdd>

  <!--  Definition of global dictionary constants          -->
  <define>
  </define>

<!--  Definition of the used visualization attributes    -->

  <display>
    <vis name="BeamPipeVis" r="0.0" g="0.9" b="0.0" showDaughters="false" visible="true"/>
    <vis name="outBeamPipeVis" r="0.9" g="0.0" b="0.0" showDaughters="false" visible="true"/>
    <vis name="paraPipeVis" r="0.0" g="0.0" b="0.9" showDaughters="false" visible="true"/>
    <vis name="IronfixVis" r="0.0" g="0.9" b="0.9" showDaughters="false" visible="true"/>
    <vis name="SteelconeVis" r="0.9" g="0.0" b="0.9" showDaughters="false" visible="true"/>
    <vis name="VacuumPipeVis" r="1.0" g="0.0" b="0.0" showDaughters="false" visible="true"/>
    <vis name="G10Vis" r="0.0" g="1.5" b="0.0" showDaughters="false" visible="true"/>
    <vis name="CuVis" r="0.0" g="0.0" b="2.0" showDaughters="false" visible="true"/>
    <vis name="ArCO2Vis" r="2.0" g="1.0" b="0.5" showDaughters="false" visible="true"/>
    <vis name="CathodeVis" r="0.0" g="2.0" b="0.5" showDaughters="false" visible="true"/>
    <vis name="FrondEndVis" r="0.0" g="0.0"  b="0.9" showDaughters="false" visible="true"/>
    <vis name="KaptonVis" r="0.9" g="0.0" b="0.0" showDaughters="false" visible="true"/>
 </display>

 <readouts>
   <readout name="trackerreadout">
      <id>system:1,x:5,y:5,z:5</id>
  </readout>
</readouts>

<detectors>
    <comment>Beam Pipe</comment>

    <detector name="Beampipe" type="SimpleCylinder" insideTrackingVolume="true" vis="BeamPipeVis">
    <dimensions rmin="BeamPipe_rmin" rmax="BeamPipe_rmax" dz="BeamPipe_Zmax" phi0="0" deltaphi="360*deg" z_offset="0*cm" material="Beryllium"/>
    </detector>

    <detector name="VaccumBeampipe" type="SimpleCylinder" insideTrackingVolume="true" vis="VacuumPipeVis">
    <dimensions rmin="0.0" rmax="BeamPipe_rmin" dz="BeamPipe_Zmax" phi0="0" deltaphi="360*deg" z_offset="0*cm" material="Vacuum" />
    </detector>

    <detector name="ParaffinPipe" type="SimpleCylinder" insideTrackingVolume="true" vis="paraPipeVis">
    <dimensions rmin="ParaPipe_rmin" rmax="ParaPipe_rmax" dz="ParaPipe_Zmax" phi0="0" deltaphi="360*deg" z_offset="0*cm" material="Steel235"/>
    </detector>

    <detector name="OuterBeamPipe" type="SimpleCylinder" insideTrackingVolume="true" vis="outBeamPipeVis">
    <dimensions rmin="OuterBeam_rmin" rmax="OuterBeam_rmax" dz="OuterBeam_Zmax" phi0="0" deltaphi="360*deg" z_offset="0*cm" material="Beryllium"/>
    </detector>
```



```
      <constant name="innerG10_rmin" value="40*mm"/>
      <constant name="innerG10_rmax" value="45*mm"/>
      <constant name="G10zmax" value="400*mm"/>

      <constant name="innerCu_rmin" value="45*mm"/>
      <constant name="innerCu_rmax" value="45.005*mm"/>

      <constant name="OuterCu_rmin" value="397.995*mm"/>
      <constant name="OuterCu_rmax" value="398*mm"/>

      <constant name="outerG10_rmin" value="398*mm"/>
      <constant name="outerG10_rmax" value="400*mm"/>

      <constant name="ArCO2_rmin" value="45.005*mm"/>
      <constant name="ArCO2_rmax" value="398*mm"/>
    <!--  <constant name="Cathode_rmax" value="173*mm"/> -->

      <constant name="kapton_rmin" value="48.005*mm"/>

 </define>

<display>
    <vis name="InvisibleNoDaughters"      showDaughters="false" visible="false"/>
    <vis name="InvisibleWithDaughters"    showDaughters="true" visible="false"/>
    <vis name="GreenVis"   alpha="1" r="0.0" g="1.0" b="0.0" showDaughters="true" visible="true"/>
    <vis name="RedVis"     alpha="1" r="1.0" g="0.0" b="0.0" showDaughters="true" visible="true"/>
    <vis name="BlueVis"    alpha="1" r="0.0" g="0.0" b="1.0" showDaughters="true" visible="true"/>
</display>


<!--<limitset name="trackerlimit">
      <limit name="step_length_max" particles="*" value="5.0" unit="mm" />
      <limit name="track_length_max" particles="*" value="5.0" unit="mm" />
      <limit name="time_max" particles="*" value="50.0" unit="ns" />
      <limit name="ekin_min" particles="*" value="0.01" unit="MeV" />
      <limit name="range_min" particles="*" value="5.0" unit="mm" />
   </limitset>-->



<include ref="BeamPipe.xml"/>
<include ref="SCT_materials.xml"/>
```

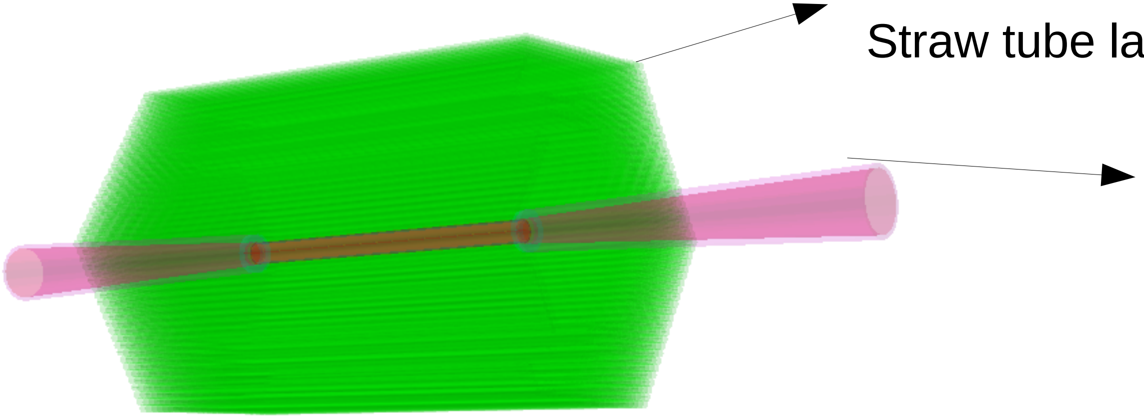
- The basic geometries like box, cone, cylinder are available in DD4hep
- The ArCO2 of TPC is specified as sensitive.



```
// DD4hep
#include "DDG4/Geant4Mapping.h"
#include "DDG4/Geant4VolumeManager.h"

// CLHEP
#include "CLHEP/Vector/ThreeVector.h"

// Geant4
#include "G4SDManager.hh"

#include "DetCommon/Geant4PreDigiTrackHit.h"

namespace det {
ArCO2SD::ArCO2SD(const std::string& aDetectorName,
                 const std::string& aReadoutName,
                 const dd4hep::Segmentation& aSeg)
    : G4VSensitiveDetector(aDetectorName), m_trackerCollection(nullptr), m_seg(aSeg) {
  // name of the collection of hits is determined byt the readout name (from XML)
  collectionName.insert(aReadoutName);
}

ArCO2SD::~ArCO2SD() {}

void ArCO2SD::Initialize(G4HCofThisEvent* aHitsCollections) {
  // create a collection of hits and add it to G4HCofThisEvent
  // deleted in ~G4Event
  m_trackerCollection = new G4THitsCollection<fcc::Geant4PreDigiTrackHit>(SensitiveDetectorName, collectionName[0]);
  aHitsCollections->AddHitsCollection(G4SDManager::GetSDMpointer()->GetCollectionID(m_trackerCollection),
                                      m_trackerCollection);
}

bool ArCO2SD::ProcessHits(G4Step* aStep, G4TouchableHistory*) {
  // check if energy was deposited
  G4double edep = aStep->GetTotalEnergyDeposit();
  if (edep == 0.) return false;
  // get track
  const G4Track* track = aStep->GetTrack();
  CLHEP::Hep3Vector prePos = aStep->GetPreStepPoint()->GetPosition();
  CLHEP::Hep3Vector postPos = aStep->GetPostStepPoint()->GetPosition();
  // create a hit and add it to collection
  // deleted in ~G4Event
  auto hit = new fcc::Geant4PreDigiTrackHit(
      track->GetTrackID(), track->GetDefinition()->GetPDGEncoding(), edep, track->GetGlobalTime());
  hit->cellID = utils::cellID(m_seg, *aStep);
  hit->prePos = prePos;
  hit->postPos = postPos;
  m_trackerCollection->insert(hit);
  return true;
}
}
```

STT geometry in DD4hep

- Geometry of Straw tube tracker is more complex than TPC.
- For the placing of straw tubes in hexagonal shape, one have to define geometry using c++
- The classes and functions available for user defined geometry can learn from ***https://dd4hep.web.cern.ch/dd4hep/reference/annotated.html***



Straw tube layers

Beam pipe

```
//
#include "DD4hep/DetFactoryHelper.h"
#include "STT/StrawTubeTracker.h"
#include "XML/XMLElements.h"
#include "XML/Layering.h"
//#include "ACTS/Plugins/DD4hepPlugins/DD4hepDetElement.hpp"
#include "ACTS/Plugins/TGeoPlugins/TGeoDetectorElement.hpp"
#include <map>





using namespace std;
using namespace dd4hep;
using namespace dd4hep::detail;

namespace det {
static Ref_t create_StrawTubeTracker(dd4hep::Detector& description, dd4hep::xml::Handle_t e, dd4hep::SensitiveDetector sens)  {
   typedef vector<dd4hep::PlacedVolume> Placements;
   xml_det_t   xmlDet        = e;
   dd4hep::Material  vacuum  = description.vacuum();
   int         det_id        = xmlDet.id();
   string      det_name      = xmlDet.nameStr();
   dd4hep::DetElement  sdet    (det_name,det_id);
   dd4hep::Assembly assembly  (det_name);
   dd4hep::Volume motherVol  = description.pickMotherVolume(sdet);
   int m_id=0, c_id=0, n_sensor=0;
   map<string, dd4hep::Volume> modules;
   map<string, Placements> sensitives;
   dd4hep::PlacedVolume pv;

   assembly.setVisAttributes(description.invisible());
   sens.setType("tracker");

   std::cout << "Test1" << std::endl;

   for (xml_coll_t mi(xmlDet,_U(module)); mi; ++mi, ++m_id)  {
      xml_comp_t x_mod = mi;
      string m_nam     = x_mod.nameStr();
      xml_comp_t tubs  = x_mod.tubs();
        double    rmin = tubs.rmin();
        double   rmax  = tubs.rmax();
        double   zhalf = tubs.zhalf();
        double   total_thickness=0.;
        xml_coll_t ci(x_mod,_U(module_component));
        for(ci.reset(), total_thickness=0.0; ci; ++ci)
        total_thickness += xml_comp_t(ci).thickness();


    std::cout << "Test2" << std::endl;
```

- The Ar-CO2 part of straw tube is specified as sensitive.
- The two geometries will be available in aurora soon

Thank you