Status of migration to DD4HEP

Sanghyun Ko

26 Jun 2020

Initialization of DD4HEP

Initializing DD4HEP for dual-readout calorimeter

- Started migration effort to DD4HEP and initialized the DD4HEP code for dual-readout calorimeter [github] (at the development branch).
- Material properties (including optical properties) are translated to xml format.
- Fortunately, the convention of geometry in DD4HEP is very similar to that of GEANT4. Expect smooth implementation of the geometry.
- However, sensitive detector, readout segmentation in DD4HEP & interface to GEANT4 are still subjects to be understood.

```
<material name="DR_Polystyrene">
  <D value="1.032" unit="g/cm3"/>
  <composite n="19" ref="C"/>
  <composite n="21" ref="H"/>
  <property name="RINDEX" ref="RI_PS"/>
  <property name="ABSLENGTH" ref="AbsLen_PS"/>
  <property name="FASTCOMPONENT" ref="scintFast_PS"/>
  <constant name="SCINTILLATIONYIELD" value="13.9/keV"/>
  <constant name="FASTTIMECONSTANT" value="2.8*ns"/>
</material>

<matrix name="scintFast_PS" coldim="2" values="
  1.37760*eV 0.
  1.45864*eV 0.
  1.54980*eV 0.
  1.65312*eV 0.
  1.71013*eV 0.
  1.77120*eV 0.
  1.83680*eV 0.
  1.90745*eV 0.0003
  1.98375*eV 0.0008
  2.06640*eV 0.0032
  2.10143*eV 0.0057
  2.13766*eV 0.0084
  2.17516*eV 0.0153
  2.21400*eV 0.0234
  2.25426*eV 0.0343
  2.29600*eV 0.0604
  2.33932*eV 0.0927
  2.38431*eV 0.1398
```

Analogous to G4Trap

to G4LogicalVolume

to G4VPhysicalVolume

```
dd4hep::Trap tower( fX_towerDim->height()/2., 0., 0., fParamBarrel->GetH1(), fParamBarrel->GetBl1(), fParamBarrel->GetTl1(), 0.,
                    fParamBarrel->GetH2(), fParamBarrel->GetBl2(), fParamBarrel->GetTl2(), 0. );

dd4hep::Volume towerVol( "tower"+std::to_string(towerNoLR), tower, fDescription->material(fX_towerDim->materialStr()) );
towerVol.setVisAttributes(*fDescription, fX_towerDim->visStr());

for (int nPhi = 0; nPhi < 30/*fX_towerDim->nphi()*/; nPhi++) {
  dd4hep::PlacedVolume towerPhys = fExperimentalHall->placeVolume( towerVol, towerNoLR*fX_towerDim->nphi()+nPhi, fParamBarrel->GetTransform3D(nPhi) );
  towerPhys.addPhysVolID("tower",towerNoLR*fX_towerDim->nphi()+nPhi);
}
```

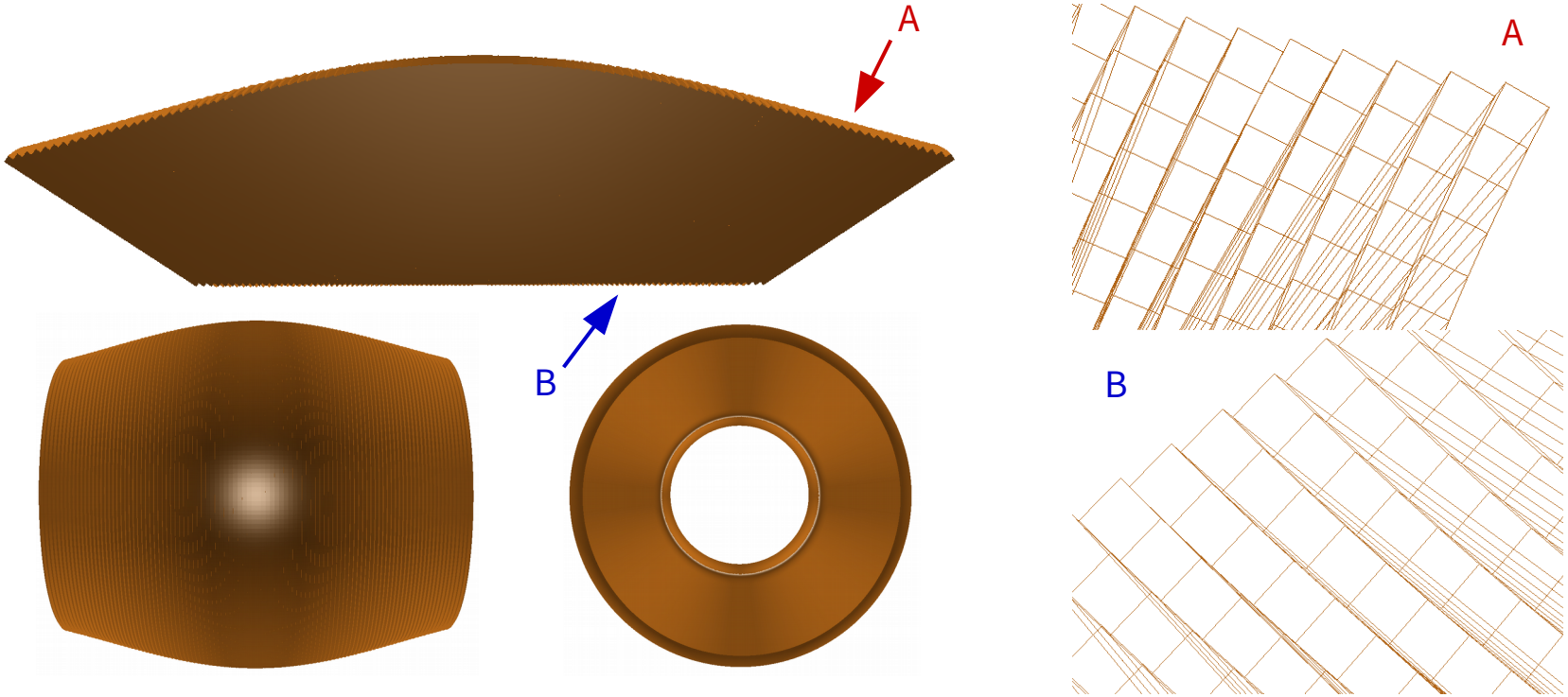
Initialization of DD4HEP

Initializing DD4HEP for dual-readout calorimeter

- Implemented barrel geometry with only copper towers.

  cf) DD4HEP seems more faster than GEANT4 for loading & visualizing geometry at first glance.

- Planning to implement (in sequence)
  - Cerenkov & scintillation fibres
  - SiPMs & Kodak filters
  - Sensitive detectors & interface to GEANT4 (similar to [1][2]) → will be able to run simulation at this stage



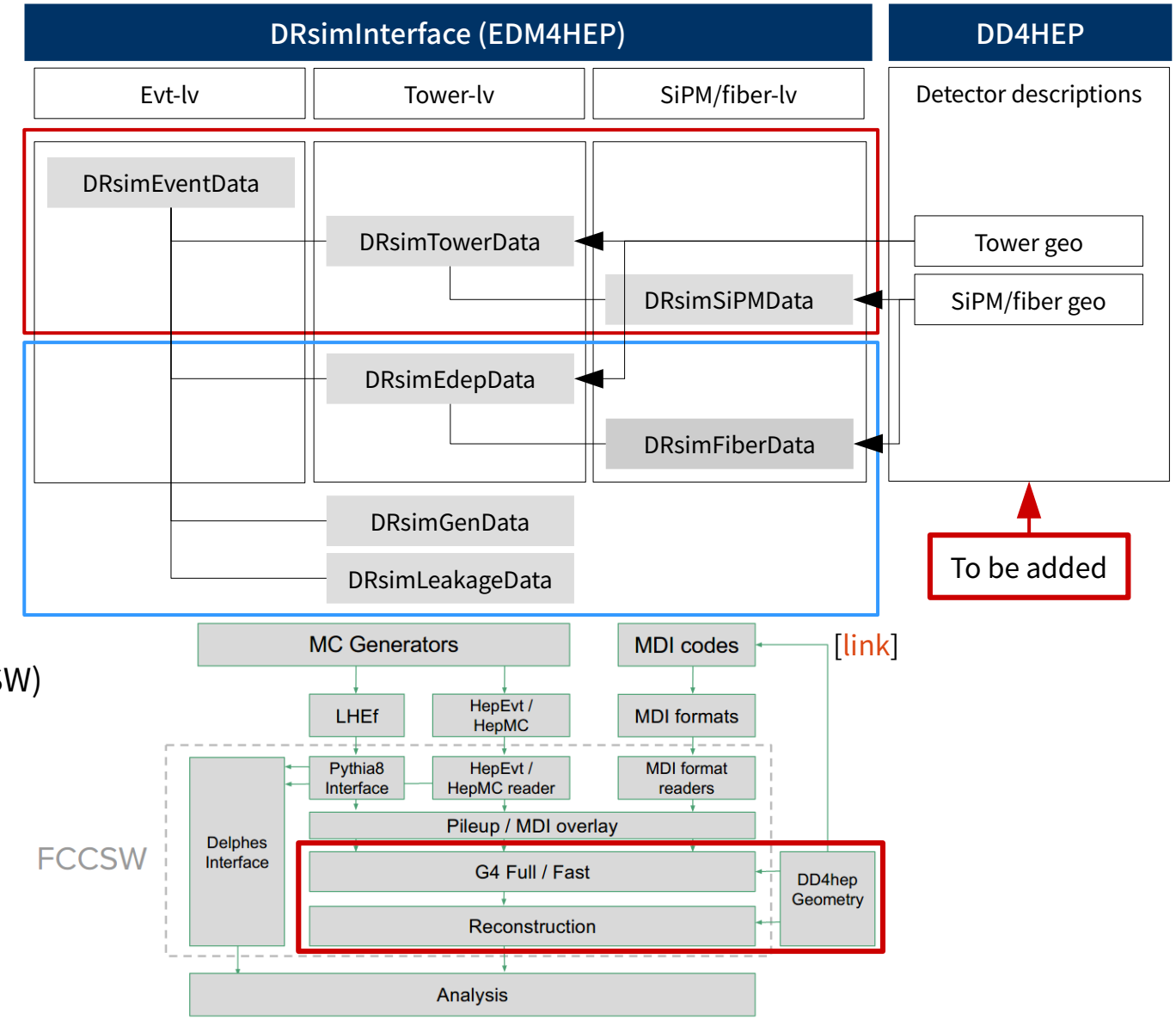
EDM with DD4HEP

Effect of DD4HEP to EDM

- Without dedicated detector description module, i.e. DD4HEP, migration to FCCSW is not possible.
- Dual-readout calorimeter has very complicated geometry with ~ O(100M) of channels.

  i.e. geometry information should not be repeated in every event.
- Instead of saving geometry information in EDM, loading detector descriptions using DD4HEP would be the ideal way.

  (as it has already been done in FCCSW)

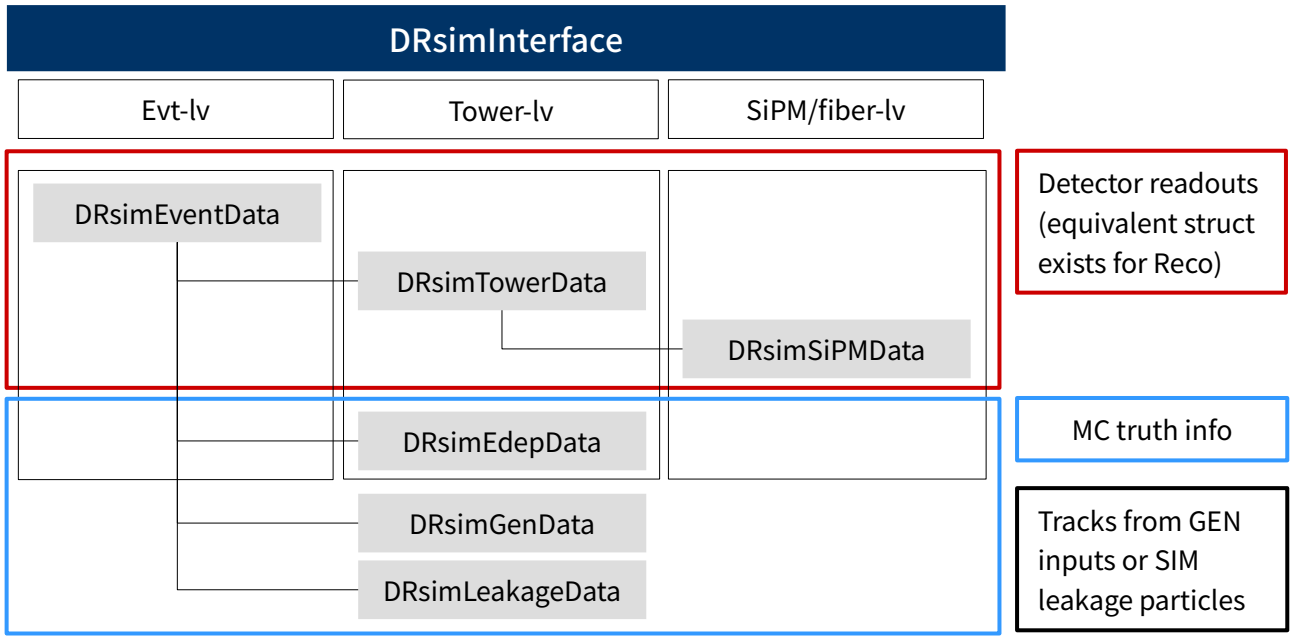


Backups

Event Data Model

Event Data Model (EDM) of dual-readout calorimeter

- At the moment, the EDM of dual-readout calorimeter simulation is constructed using the C++ class & structures [Github].
- EDM is designed according to the geometrical hierarchy of dual-readout calorimeter.
- I/O of a file is based on the TTree of ROOT [Github].
- Costs O(10 kb) – O(100 kb) per events.
- Expect to be easily converted into existing EDM formats of FCCSW.

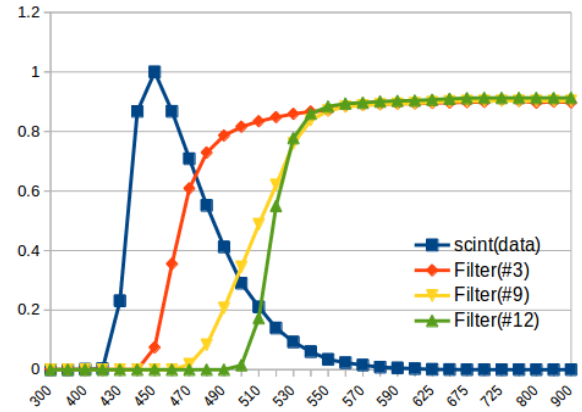



```
class DRsimInterface {
public:
  DRsimInterface();
  ~DRsimInterface();

  typedef std::pair<float,float> hitRange;
  typedef std::pair<int,int> hitXY;
  typedef std::map<hitRange, int> DRsimTimeStruct;
  typedef std::map<hitRange, int> DRsimWavlenSpectrum;
  typedef std::tuple<float,float,float> threeVector;

  struct DRsimSiPMData {
    DRsimSiPMData() {};
    virtual ~DRsimSiPMData() {};

    int count;
    int SiPMnum;
    int x;
    int y;
    threeVector pos;
    DRsimTimeStruct timeStruct;
    DRsimWavlenSpectrum wavlenSpectrum;
  };

  struct DRsimTowerData {
    DRsimTowerData() {};
    virtual ~DRsimTowerData() {};

    std::pair<int,float> towerTheta;
    std::pair<int,float> towerPhi;
    int numx;
    int numy;
    float innerR;
    float towerH;
    float dTheta;
    std::vector<DRsimSiPMData> SiPMs;
  };
```

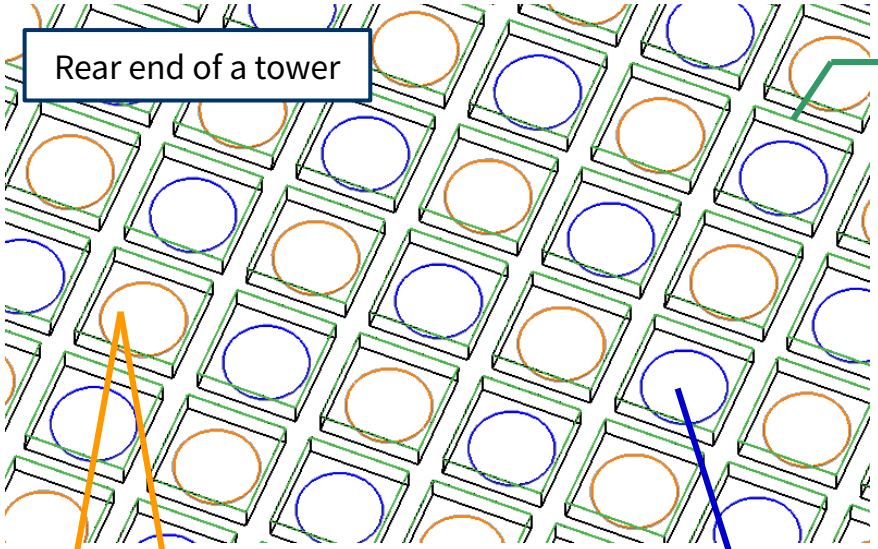
Required optical properties for DD4HEP

GEANT4 setup – required optical properties for the simulation [Github]

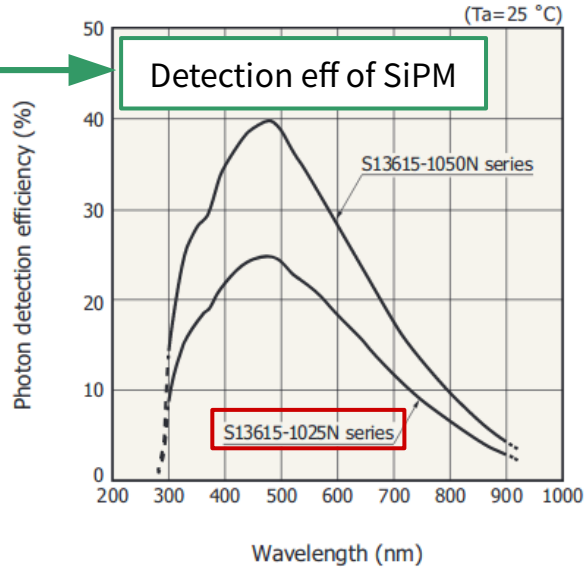
Transmission eff of filters



Rear end of a tower



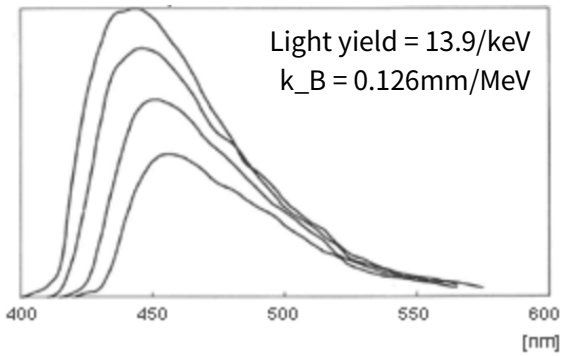
Detection eff of SiPM



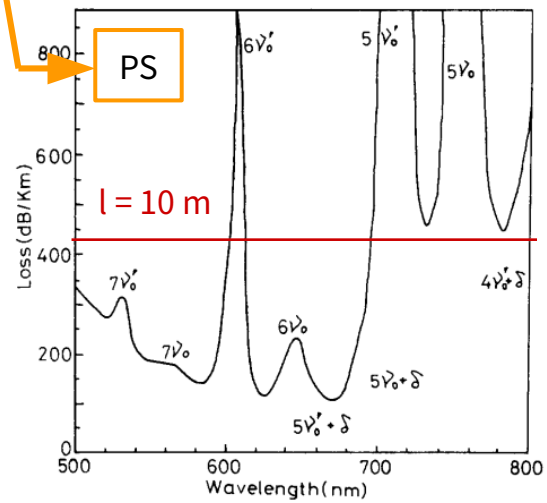
Attenuation loss diverges at 400nm → applied filter to S channel to mitigate it

Attenuation loss of Polystyrene (PS) & PMMA

Scintillation spectra of PS



PS



PMMA