LArG4 refactoring status

Hans Wenzel

10th April 2018

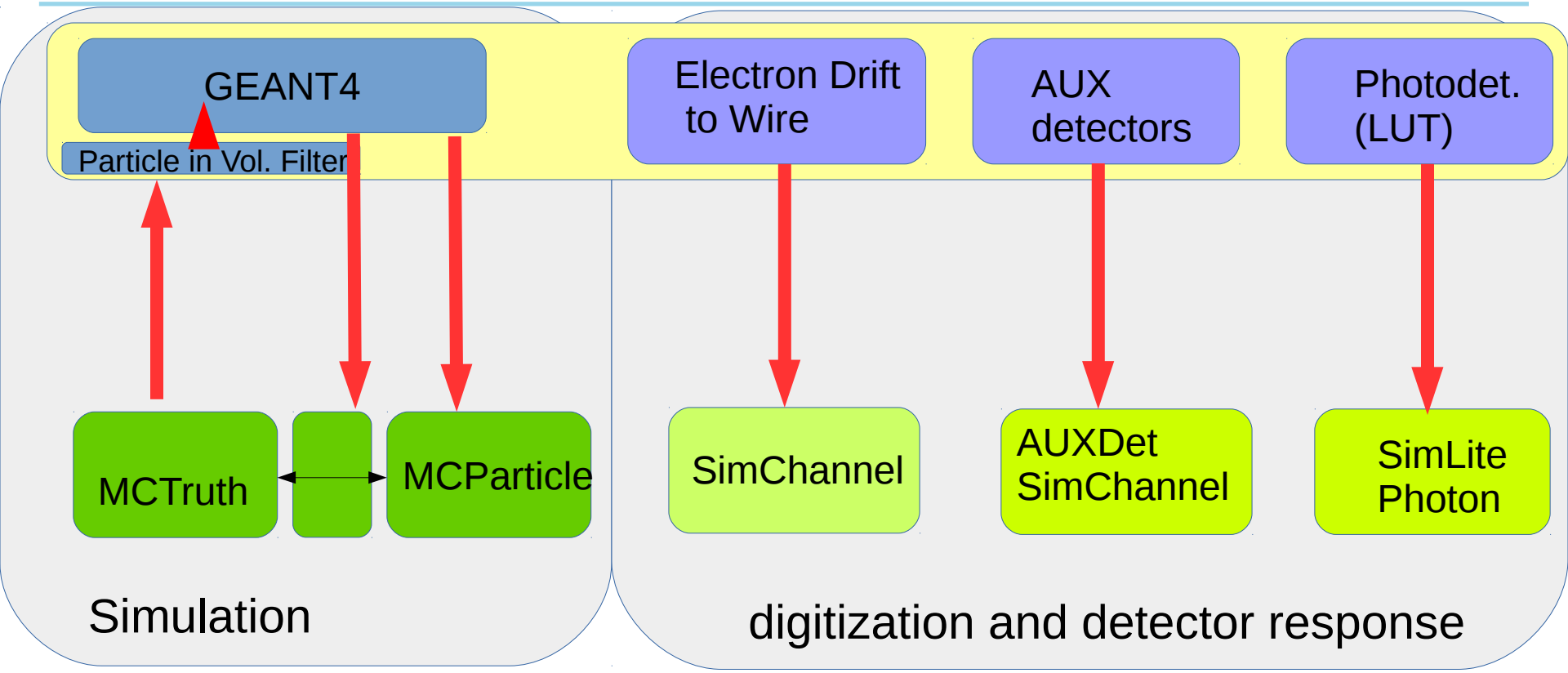
Outline

- Requirements
- What does LArG4 do?
- How to refactor it status thereof/ how to access artg4tk
- Selection of physics list, processes etc. fcl parameters
- artg4tk how to run it.
- How to define a detector in gdml.
- Some results

Requirements

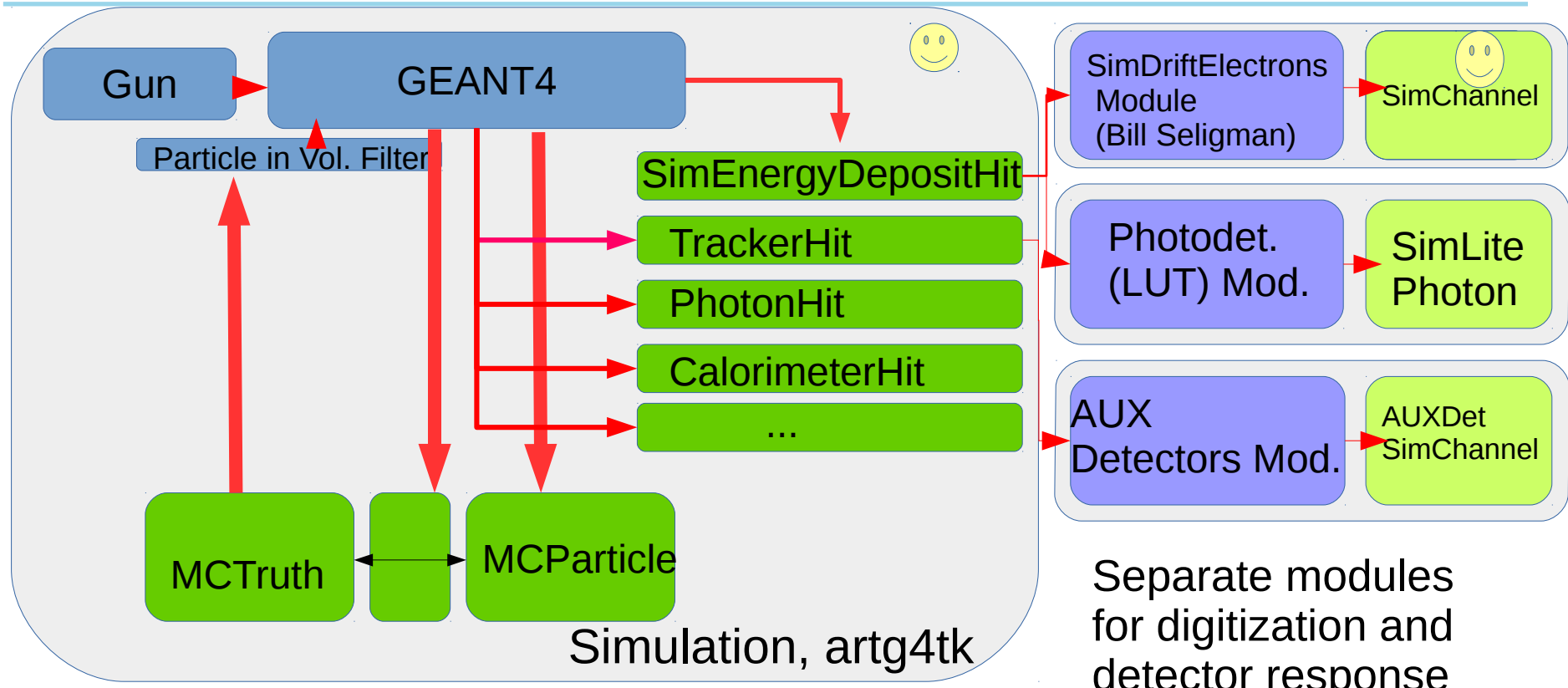
- Separate Simulation from digitization and detector response
- Simulation:
  - completely depend on tools provided by Geant4 and use the provided interfaces (or work with the Geant4 collaboration to make them available), make sure that they are efficient with regards to CPU and memory (profiling).
  - Have access to all physics lists and physics constructors and processes provided by GEANT4
  - Be able to describe complete detector systems → liquid Argon TPC is just one possible sensitive Detector.
  - Read out optical Scintillation photons from the liquid Argon surrounding the TPC.
- digitization and detector response:
  - Ability to plug in models handling the correlations between ionization and Scintillation (e.g. Nest)
  - Ability to switch drift model.
-

LArG4_module



- Monolithic hard to work with
- Mixes digitization and detector response with simulation
- Takes routines out of Geant4 and modifies them instead of using standard interfaces
- Physics processes incomplete, stuck with restrictions that existed at the time...
- Binds processes to specific material/volume
- Read out photons only in TPC volume
- ...

Refactored



Separate modules for digitization and detector response

feature/seligman_DriftIonizationElectrons routine larsim/DetSim/SimDriftElectrons_module.cc

https://cdcvs.fnal.gov/redmine/projects/artg4tk

https://cdcvs.fnal.gov/redmine/projects/g4mps/wiki/Phase2_App_01052016#Download-and-install-external-tools-and-build-artg4tk-on-a-generic-node

Physics

Use the new physics list factory:

- Available via nusoft
- Access to all reference physics list
- Extendable to add all the available geant4 physics constructors e.g we use (parameters controlled via fcl):
  - Optical physics
  - Steplimiter for charged particles in active TPC volume.
  - Time limiter for neutrons.
  - More precise em physics can be selected.
  -

Configure the Physics in the fcl file

```
    DetectorHolder: {}
    ActionHolder: {}
    RandomNumberGenerator: {}
    PhysicsListHolder: {}
    PhysicsList: {
        PhysicsListName: "FTFP_BERT"
        DumpList: false
        enableCerenkov: false
        enableScintillation: true
        ScintillationByParticleType: false
        enableAbsorption: false
        enableRayleigh: false
        enableMieHG: false
        enableBoundary: false
        enableWLS: false
}

    // Detector(s) for the simulation
    GDMLDetector :
    {
    category: "world"
    gdmlFileName_ : "lArDet.gdml"
    }
```

Physics constructor

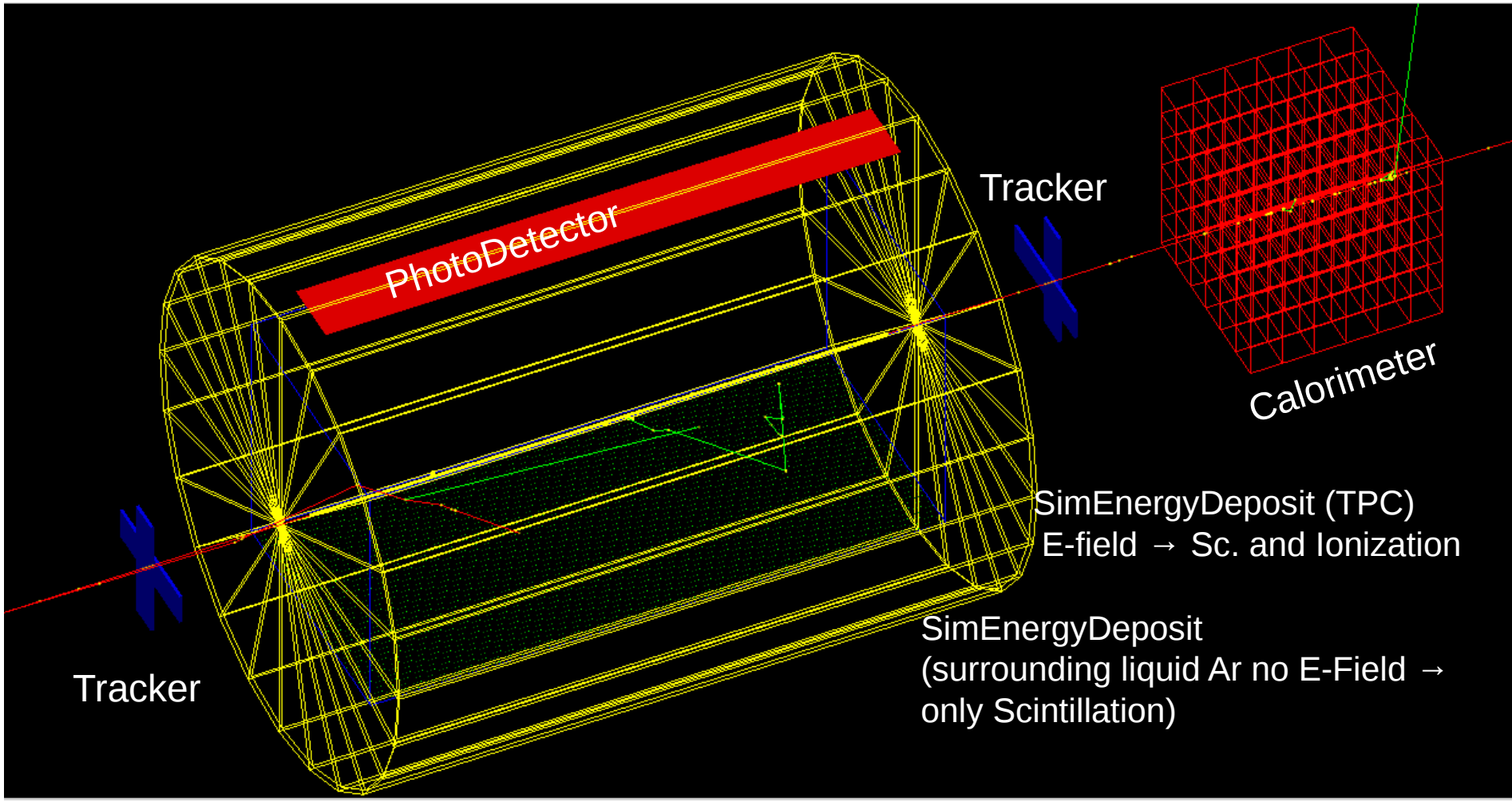
```
artg4tk::PhysicsListService::PhysicsListService(fhicl::ParameterSet const & p, art::ActivityRegistry &) :
  PhysicsListName_( p.get<std::string>("PhysicsListName","FTFP_BERT")),
  DumpList_( p.get<bool>("DumpList",true)),
  enableNeutronLimit_(p.get<bool>("enableNeutronLimit",true)),
  NeutronTimeLimit_(p.get<double>("NeutronTimeLimit",10.*microsecond)),
  NeutronKinELimit_(p.get<double>("NeutronKinELimit",0.0)),
  enableStepLimit_(p.get<bool>("enableStepLimit",true)),
  enableOptical_(p.get<bool>("enableOptical",true)),
  enableCerenkov_( p.get<bool>("enableCerenkov",false)),
  CerenkovStackPhotons_( p.get<bool>("CerenkovStackPhotons",false)),
  CerenkovMaxNumPhotons_(p.get<int>(" CerenkovMaxNumPhotons",100)),
  CerenkovMaxBetaChange_(p.get<double>("CerenkovMaxBetaChange",10.0)),
  CerenkovTrackSecondariesFirst_( p.get<bool>("ScintillationTrackSecondariesFirst",false)),
  enableScintillation_( p.get<bool>("enableScintillation",true)),
  ScintillationStackPhotons_( p.get<bool>("ScintillationStackPhotons",false)),
  ScintillationByParticleType_( p.get<bool>("ScintillationByParticleType",false)),
  ScintillationTrackInfo_( p.get<bool>("ScintillationTrackInfo",false)),
  ScintillationTrackSecondariesFirst_( p.get<bool>("ScintillationTrackSecondariesFirst",false)),
  enableAbsorption_( p.get<bool>("enableAbsorption",false)),
  enableRayleigh_( p.get<bool>("enableRayleigh",false)),
  enableMieHG_( p.get<bool>("enableMieHG",false)),
  enableBoundary_( p.get<bool>("enableBoundary",false)),
  enableWLS_( p.get<bool>("enableWLS",false)),
  BoundaryInvokeSD_( p.get<bool>("BoundaryInvokeSD",false)),
  WLSProfile_( p.get<std::string>(" WLSProfile","delta"))
{}
```

```
G4VUserPhysicsList* artg4tk::PhysicsListService::makePhysicsList() {

  g4alt::G4PhysListFactory factory;
    // Access to registries and factories
    //
  G4PhysicsConstructorRegistry* g4pcr = G4PhysicsConstructorRegistry::Instance();
  G4PhysListRegistry* g4plr = G4PhysListRegistry::Instance();
  //
  // the following should be unneccessary at some point:
  //
  g4plr->AddPhysicsExtension("OPTICAL", "G4OpticalPhysics");
  g4plr->AddPhysicsExtension("STEPLIMIT", "G4StepLimiterPhysics");
  g4plr->AddPhysicsExtension("NEUTRONLIMIT", "G4NeutronTrackingCut");
  g4pcr->PrintAvailablePhysicsConstructors();
  g4plr->PrintAvailablePhysLists();
  G4VModularPhysicsList* phys = NULL;
  G4String physName = PhysicsListName_;
  if (enableOptical_)   physName=physName+"+OPTICAL";
  if (enableStepLimit_) physName=physName+"+STEPLIMIT";
  if (enableNeutronLimit_)  physName=physName+"+NEUTRONLIMIT";
  std::cout << " Name of Physics list: "<< physName<<std::endl;
  if (factory.IsReferencePhysList(physName)) {
    phys = factory.GetReferencePhysList(physName);
  }
    std::cout << phys->GetPhysicsTableDirectory() << std::endl;
    if (enableOptical_)
      {
                G4OpticalPhysics* opticalPhysics = (G4OpticalPhysics*) phys->GetPhysics("Optical");
                opticalPhysics->Configure(kCerenkov,  enableCerenkov_);
                opticalPhysics->SetCerenkovStackPhotons(CerenkovStackPhotons_);
                opticalPhysics->Configure(kScintillation,  enableScintillation_);
                opticalPhysics->SetScintillationStackPhotons(ScintillationStackPhotons_);
                opticalPhysics->SetScintillationByParticleType(ScintillationByParticleType_);
                opticalPhysics->SetScintillationTrackInfo(ScintillationTrackInfo_);
                opticalPhysics->SetTrackSecondariesFirst(kCerenkov, true);      // only relevant if we actually stack and trace the optical photons
                opticalPhysics->SetTrackSecondariesFirst(kScintillation, true); // only relevant if we actually stack and trace the optical photons
                opticalPhysics->SetMaxNumPhotonsPerStep(CerenkovMaxNumPhotons_);
                opticalPhysics->SetMaxBetaChangePerStep( CerenkovMaxBetaChange_);
                opticalPhysics->Configure(kAbsorption,enableAbsorption_);
                opticalPhysics->Configure(kRayleigh,enableRayleigh_);
                opticalPhysics->Configure(kMieHG,enableMieHG_);
                opticalPhysics->Configure(kBoundary,enableBoundary_);
                opticalPhysics->Configure(kWLS,enableWLS_);
      }
    if (enableNeutronLimit_)
      {
                G4NeutronTrackingCut * neutrcut = (G4NeutronTrackingCut*) phys->GetPhysics("neutronTrackingCut");
                neutrcut->SetTimeLimit(NeutronTimeLimit_);
      }
    if (DumpList_)
      {
                phys->DumpList();
                phys->DumpCutValuesTable();
      }
    return phys;
}

using artg4tk::PhysicsListService;
```

GDML(+ extensions): a complete description of detector configuration (at runtime)

- Materials, volumes etc….
- Assign step-limits to specific volumes.
- Optical properties (bulk and surface)
- Assignment of sensitive detectors of predefined type to logical volumes→ automatically trigger the creation and filling of the appropriate hit collections
- Assignment of optical surfaces
- Visualization attributes (color, solid,….)
- Makes use of formulas and loops to keep gdml file compact
- Homogeneous electric field (no electric field→ no separation of charge)
-

Example:
https://cdcvs.fnal.gov/redmine/projects/artg4tk/repository/revisions/develop/entry/gdml/lArDet.gdml

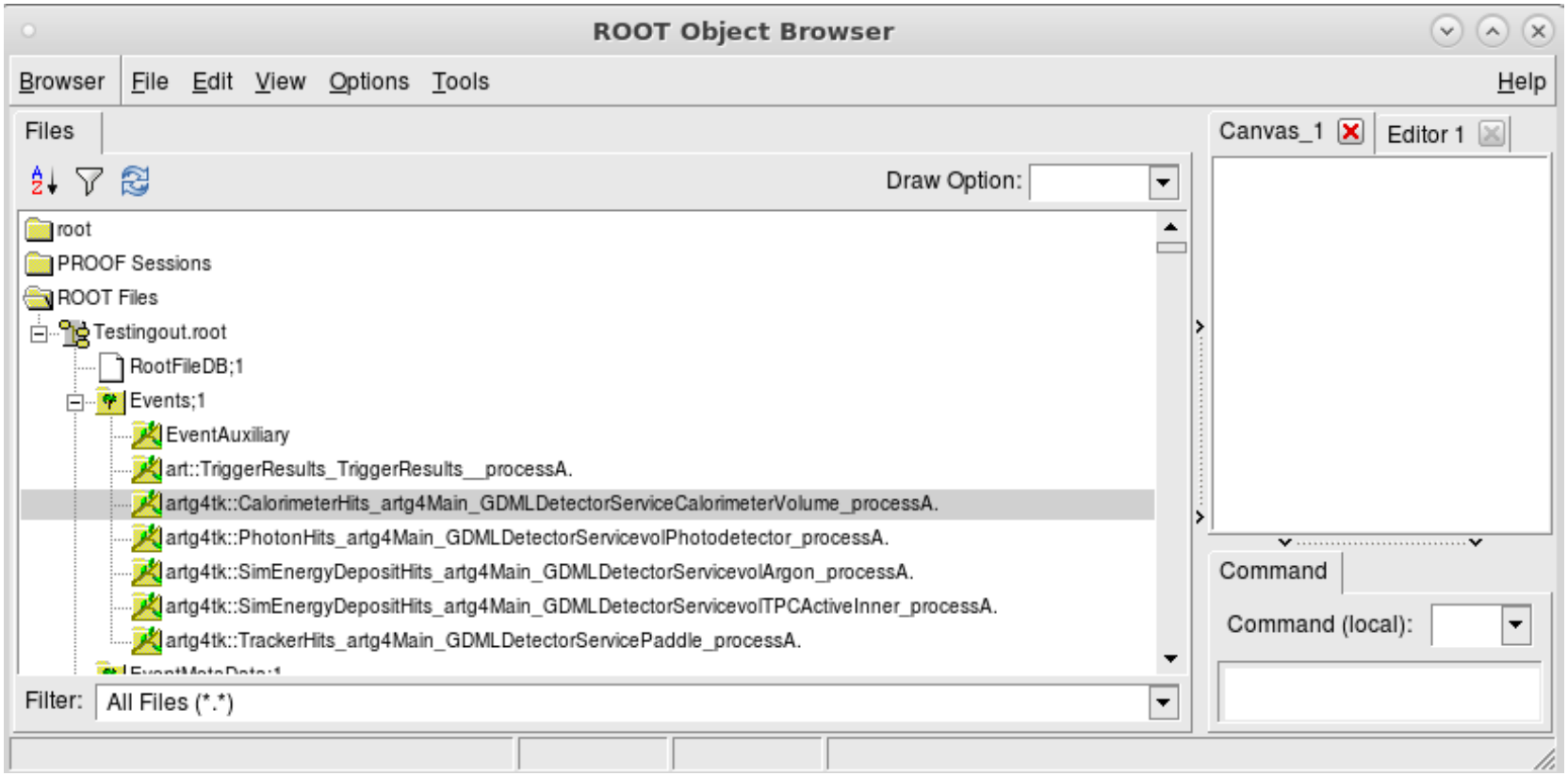


Assigning Sensitive detector to a volume:

```
<volume name="volTPCActiveInner">
    <materialref ref="LAr"/>
    <solidref ref="TPCVolume"/>
    <auxiliary auxtype="SensDet" auxvalue="SimEnergyDeposit"/>
    <auxiliary auxtype="Color" auxvalue="Blue"/>
    <auxiliary auxtype="StepLimit" auxvalue="0.01"/>
    <auxiliary auxtype="Efield" auxvalue="1000."/>
    <loop for="i" from="0" to="num" step="1">
        <physvol name="psenseWireVolume">
            <volumeref ref="SenseWire"/>
            <position name="posijk"  unit="mm" x="-200.0+(i+1)*5." y="-199.8" z="0"/>
        </physvol>
    </loop>
</volume>
<volume name="volPhotodetector">
    <materialref ref="Silicon"/>
    <solidref ref="PhotoBox"/>
    <auxiliary auxtype="SensDet" auxvalue="PhotonDetector"/>
    <auxiliary auxtype="Color" auxvalue="Red"/>
    <auxiliary auxtype="Solid" auxvalue="True"/>
</volume>
<volume name="volArgon">
    <materialref ref="LAr"/>
    <solidref ref="ArgonVolume"/>
    <auxiliary auxtype="SensDet" auxvalue="SimEnergyDeposit"/>
    <auxiliary auxtype="Color" auxvalue="Yellow"/>
    <physvol name="pCalorimeterVolume">
        <volumeref ref="volTPCActiveInner"/>
        <position name="Calpos" x="0" y="0" z="0"/>
    </physvol>
    <physvol name="pvolPhotodetector">
        <volumeref ref="volPhotodetector"/>
        <position name="photondetectorpos" unit="mm" x="0" y="391." z="0"/>
    </physvol>
</volume>
```

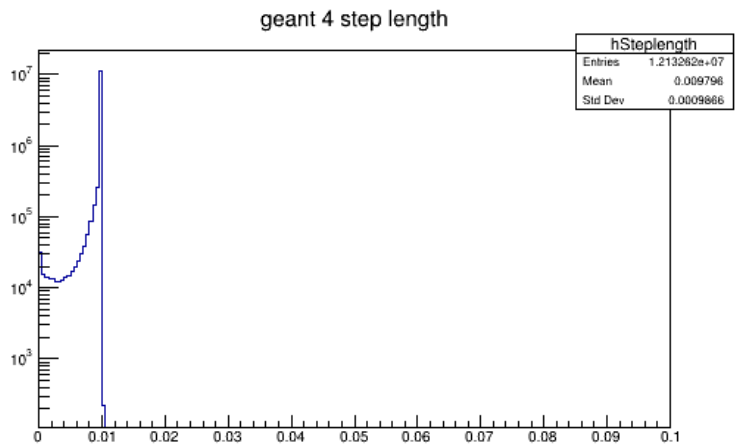
Running artg4tk and Resulting Hit Collections in the EDM

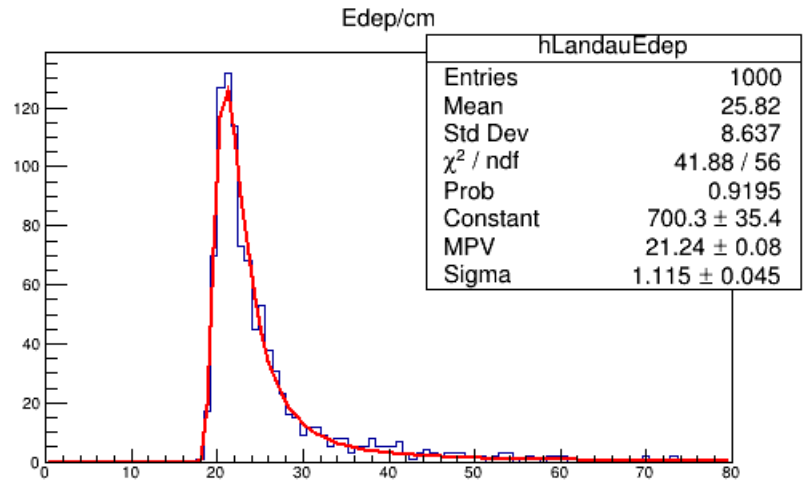
art -c lArDet.fcl

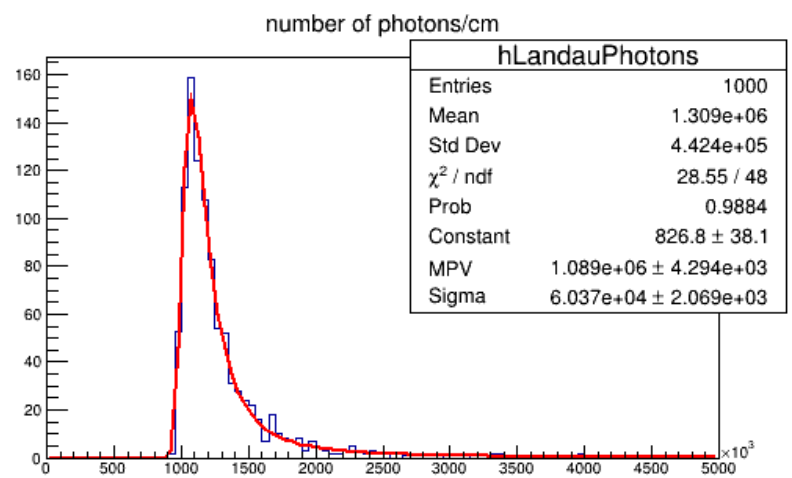


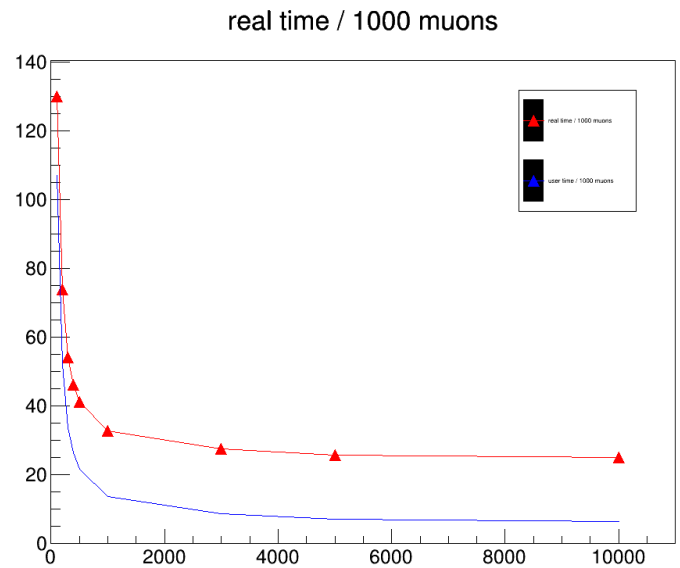
```
process_name:processA
source: {
  module_type: EmptyEvent
  maxEvents:  10
}
services: {
  message : {
     debugModules : ["*"]
     suppressInfo : []
     destinations : {
       LogToConsole : {
         type : "cout"
         threshold : "DEBUG"
         categories : {
           default : { limit : 50 }
         }
       }
     }
  }
  TFileService :
  {
    fileName      : "CheckSimEnergyDepositHits.root"
  }

    DetectorHolder: {}
    ActionHolder: {}
    RandomNumberGenerator: {}
    PhysicsListHolder: {}
    PhysicsList: {
        PhysicsListName: "FTFP_BERT"
        DumpList: false
        enableCerenkov: false
        enableScintillation: true
        ScintillationByParticleType: false
        enableAbsorption: false
        enableRayleigh: false
        enableMieHG: false
        enableBoundary: false
        enableWLS: false
}

    // Detector(s) for the simulation
    GDMLDetector :
    {
    category: "world"
    gdmlFileName_ : "lArDet.gdml"
    }


    ExampleGeneralAction: {
      name: "exampleGeneral"
    }
      myParticleGunAction: {
      name: "myParticleGun"
      NParticle: 1
      Name: "mu+"
      Direction: [ 0,  0,  1  ]
      Energy: 10.
      Position: [ 0,  0,  -130.  ]
    }
}

outputs: {
out1: {
  module_type: RootOutput
  fileName: "Testingout.root"
  }
}

physics: {
  producers: {
    artg4Main: {
      module_type: artg4Main
      enableVisualization: false
      macroPath: ".:./macros"
      visMacro: "vis.mac"
      //afterEvent: pause
    }

  }
  analyzers: {
   CheckSimEnergyDepositHit: {   module_type: CheckSimEnergyDepositHit
                    hist_dir: "HistoDir"}
   CheckPhotonHits: {   module_type: CheckPhotonHits
                    DumpGDML: true}
}

  path1: [ artg4Main ]
  stream1:  [ out1,CheckSimEnergyDepositHit ,CheckPhotonHits  ]

  trigger_paths: [ path1 ]
  end_paths: [ stream1 ]
}
```

Fermilab


geant 4 step length
hSteplength
Entries 1.213262e+07
Mean 0.009796
Std Dev 0.0009866


Edep/cm
hLandauEdep
Entries 1000
Mean 25.82
Std Dev 8.637
χ² / ndf 41.88 / 56
Prob 0.9195
Constant 700.3 ± 35.4
MPV 21.24 ± 0.08
Sigma 1.115 ± 0.045


number of photons/cm
hLandauPhotons
Entries 1000
Mean 1.309e+06
Std Dev 4.424e+05
χ² / ndf 28.55 / 48
Prob 0.9884
Constant 826.8 ± 38.1
MPV 1.089e+06 ± 4.294e+03
Sigma 6.037e+04 ± 2.069e+03


real time / 1000 muons

steplength