Simulation and analysis framework for IR3 test

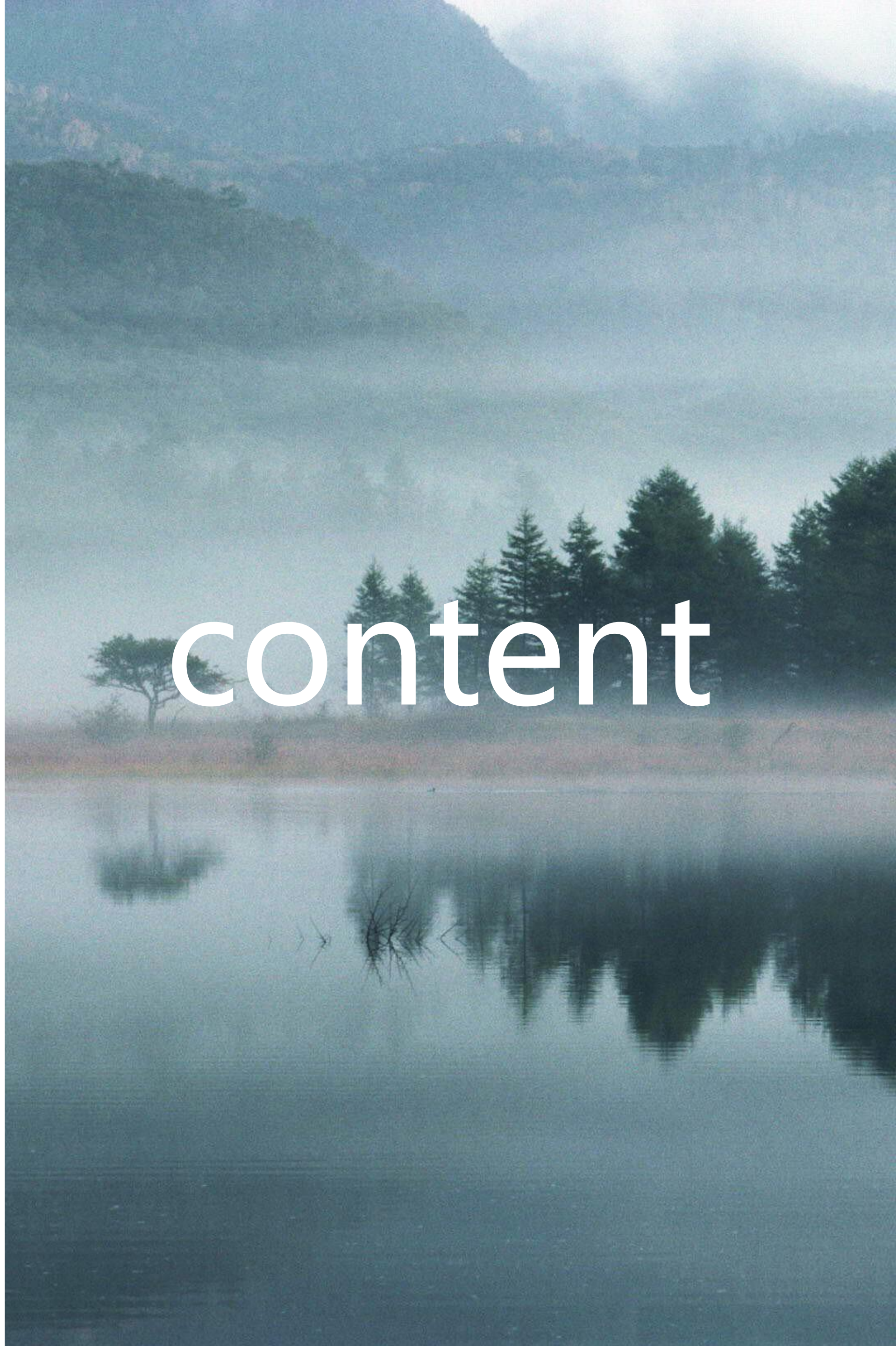
Han Miao

University of Chinese Academy of Sciences (UCAS)

Institute of High Energy Physics (IHEP)

2022.9.26

2nd Workshop on electromagnetic dipole moments of unstable particles



content

1 Simulation setup

2 Implementation of channeling process

3 Analysis framework

4 Summary and next work

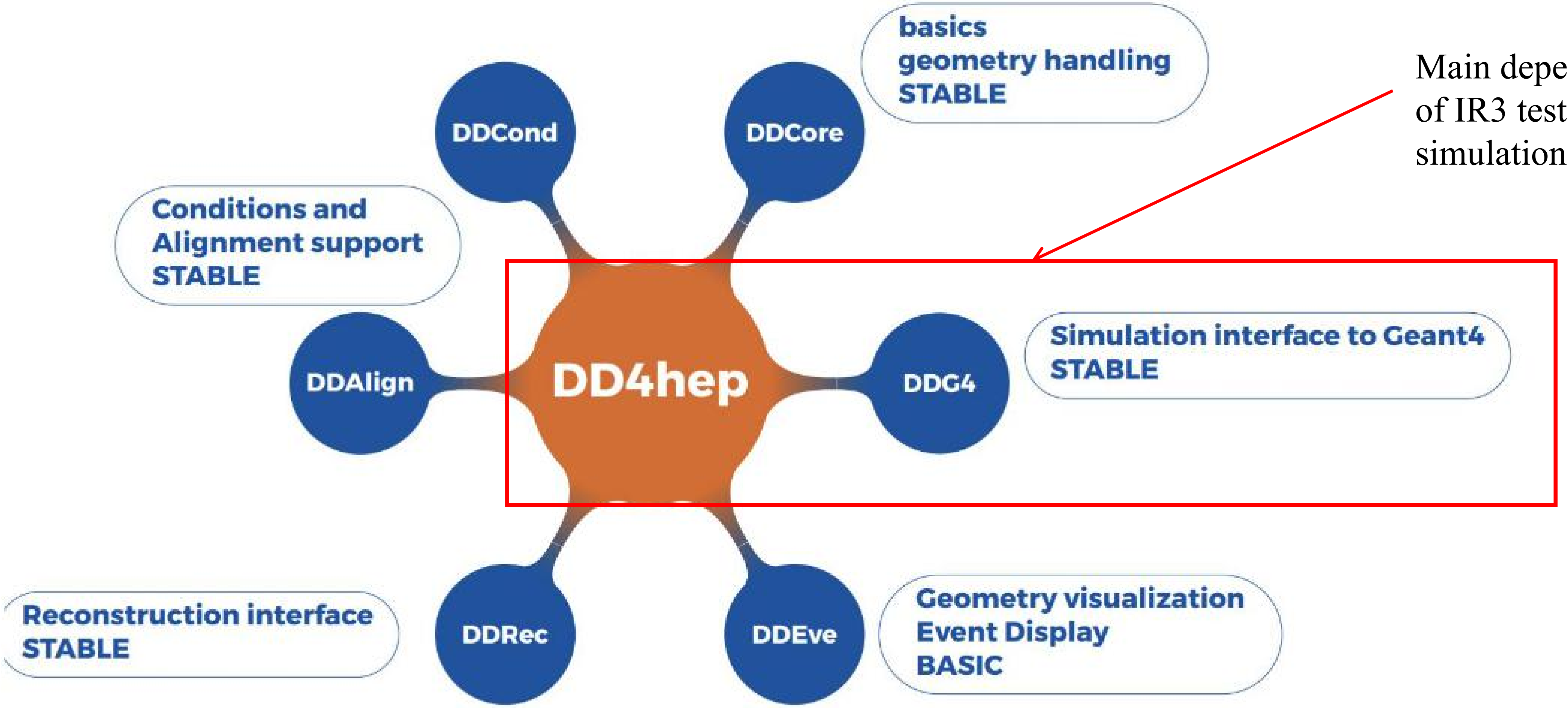
One

Simulation setup

detector goemetry, visualization, generator

DD4hep

Simulation framework for IR3 test is developed based on DD4hep (https://github.com/AIDASoft/DD4hep)



Main dependency of IR3 test simulation

Fig. 1 Structure of DD4hep

Geometry

- Simulation is configured by `xml` files
- Geometry (according to Elisabetta Spadaro Norella):
  - Target: W, 2 cm long
  - Crystal2: Si, 7 cm long, 7 mrad
  - Beam pipe: Cu OFE, elliptical form
  - MCBW Magnet: Fe, at 1 m from crystal
    - ✓ B=1.1 T, L=1.7 m
    - ✓ Bore: $R_B(x, y) = (2.6, 7.2)$ cm
  - Tracker stations: 2 blocks of 4 trackers before and after magnet
    - ✓ Si, 300 μm thick, 15×15 cm^2
    - ✓ Tracker block length=40 cm
  - (Transition radiation detector (TRD))

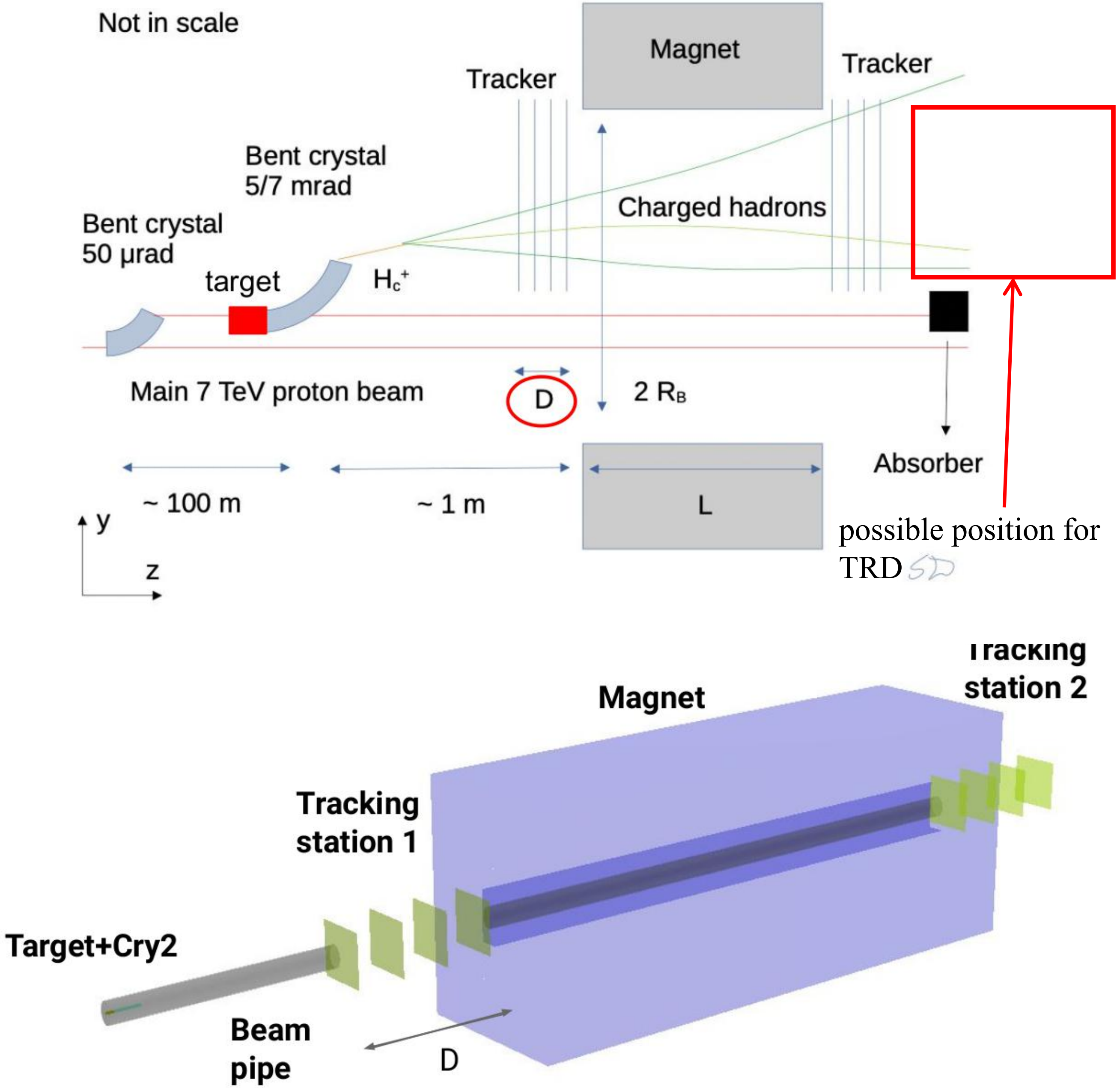


Fig. 2 Upper: A schematic picture of IR3 test; Lower: Geometry in simulation

Generator

- **Simple particle gun** (`Geant4ParticleGun`)
  - generate a monoenergetic beam of a certain kind of particle, for example, 1 TeV K, π or p

- **Input file generated by Pythia/Angantyr** (`Geant4InputAction`)

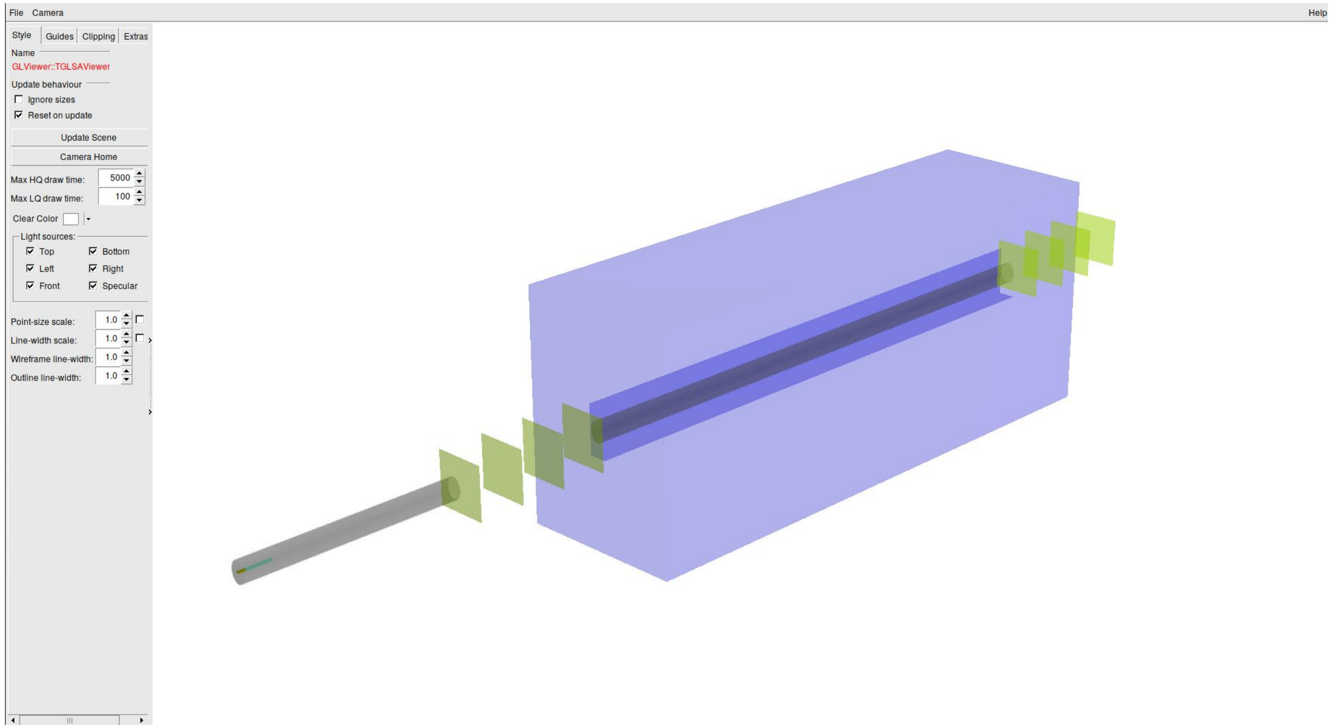
- **General particle source from Geant4** (`Geant4GPS`) (not usable yet)
  - allow users to define several different particle sources with different shapes, angular distributions and spectra. Will benefit a lot for studying special cases and backgrounds

- Smearing of vertex (by Elisabetta Spadaro Norella) (`Geant4InteractionVertexSmear`)
  - Gaussiain distribution with σ=4 mm in both x and y direction
  - The density along z axis is calculated by the interaction length in W target.

Visualization

- **geoDisplay implemented in DD4hep**
  - ➢ Only geometry will be displayed
- **Visualization rendered by OpenGL with UI by Qt5 (Geant4)**
  - ➢ Trajectories will be displayed
  - ➢ Not suggested for simulating large quantity of events or complex events
- **DDEve implementation** (not usable yet)
  - ➢ Abundant information including hits, tracks
  - ➢ Not able to display real-time
- **JavaScript ROOT (JSROOT)** (under consideration)
  - Interactive graphics in the web browsers for most of the ROOT classes including TGeometry
  - Users are able to display an event by the browser locally or remotely

Visualization



Name: GLViewer::TGLSAViewer

Update behaviour
- □ Ignore sizes
- ☑ Reset on update

Max HQ draw time: 5000

Max LQ draw time: 100

Clear Color □

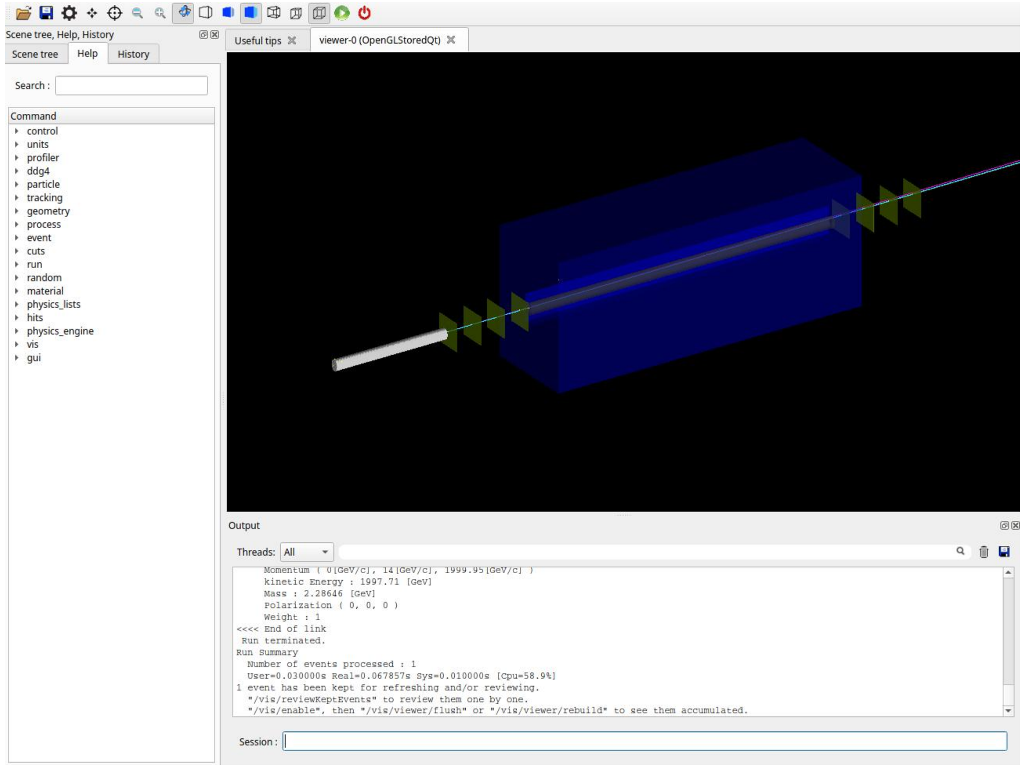
Light sources:
- ☑ Top
- ☑ Bottom
- ☑ Left
- ☑ Right
- ☑ Front
- ☑ Specular

Point-size scale: 1.0 □

Line-width scale: 1.0 □

Wireframe line-width: 1.0

Outline line-width: 1.0



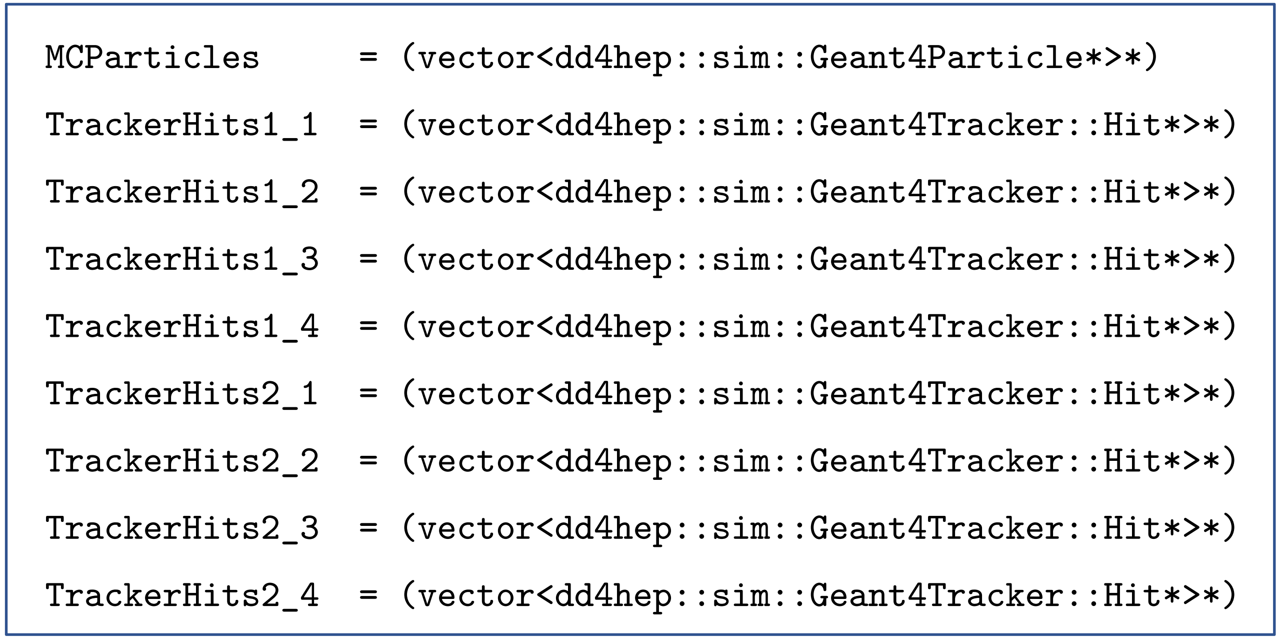
Threads: All

```
     Momentum ( 0[GeV/c], 14[GeV/c], 1999.95[GeV/c] )
     kinetic Energy : 1997.71 [GeV]
     Mass : 2.28646 [GeV]
     Polarization ( 0, 0, 0 )
     Weight : 1
<<<< End of link
 Run terminated.
Run Summary
  Number of events processed : 1
  User=0.030000s Real=0.067857s Sys=0.010000s [Cpu=58.9%]
1 event has been kept for refreshing and/or reviewing.
  "/vis/reviewKeptEvents" to review them one by one.
  "/vis/enable", then "/vis/viewer/flush" or "/vis/viewer/rebuild" to see them accumulated.
```

Fig. 3 Left: Geometry visualization by DD4hep (geoDisplay); Right: Event visualization by Geant4

Event model

- The simple event model in DDG4 is used now
  - ➢ Hits of each tracker is recorded in a tree of ROOT
  - ➢ Truth information is recorded independently in a branch
- Future plan:
  - ➢ EDM4hep may be used
    - ✓ Widely used in HEP experiments like CEPC
    - ✓ A generic event model for all procedures including digitization, reconstruction and analysis
  - ➢ Event model developed independently
    - ✓ Specialized for the requirements or IR3 test and future experiments



```
MCParticles      = (vector<dd4hep::sim::Geant4Particle*>*)
TrackerHits1_1   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits1_2   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits1_3   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits1_4   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits2_1   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits2_2   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits2_3   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
TrackerHits2_4   = (vector<dd4hep::sim::Geant4Tracker::Hit*>*)
```

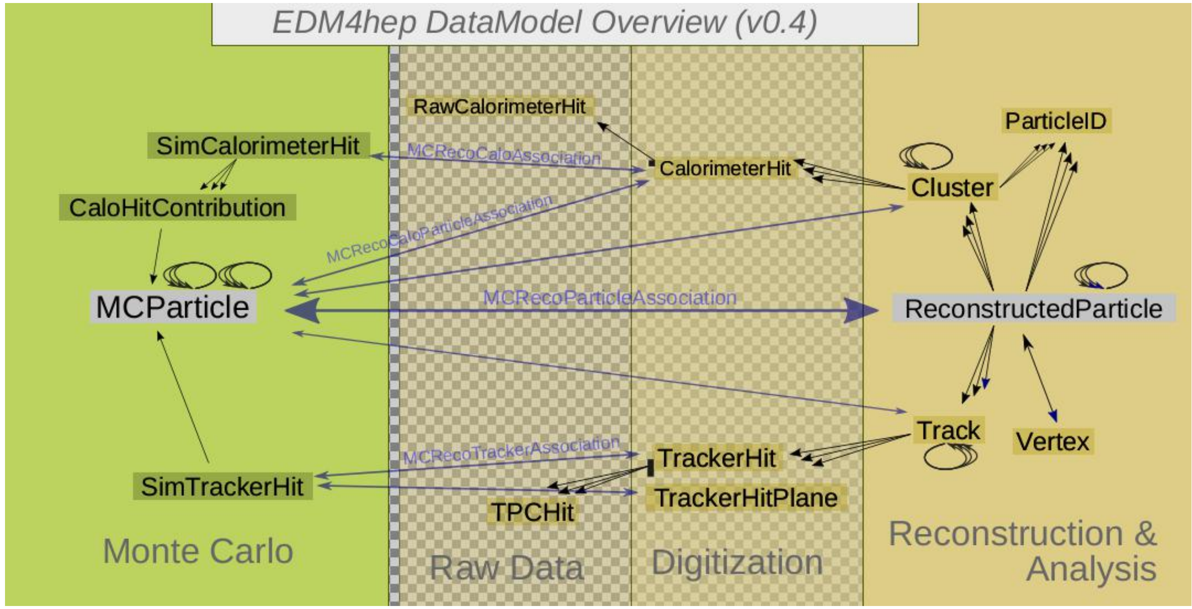


Fig. 4 Structure of EDM4hep

Physics model

- `FTFP_BERT` is used now
  - Bertini intranuclear cascade is responsible for the inelastic hadron-nucleus processes of most hadrons over the range [0, 6] GeV
  - Fritiof parton model (FTF) is used over 3 GeV to 100 TeV

- `QGSP_BERT` is under consideration
  - Inelastic processes will be handled by the Quark-gluon String model (QGS) above 12 GeV

- Further investigation are necessary to select a physics list more suitable for our case

Two

Implementation of channeling process

channeling process

Channeling process

by Chiara Maccani

- Spin of particles will be rotated in bent crystals. The rotation angle will be determined by the anomalous magnetic moment μ’.
- Already implemented into Geant4 (`G4ChannelingPhysicsList`)
- Some extended features in Geant4 needed but not implemented to DD4hep before
  - ➢ `G4CrystalExtension`: info on geometry, elasticity, ..(`G4CrystalUnitCell, G4CrystalAtomBase, G4AtomicBond`)
  - ➢ `G4ChannelingMaterialData`: info on crystal electric field, potential, nuclei and electron density, bending..(`G4ChannelingECHARM`)
  - ➢ All above information passed to `G4LogicalCrystalVolume`
- Channeling process can be simulated since new version of DD4hep
- Properties are set in `xml` files the same as other parameters

Channeling process

by Chiara Maccani

- Example in Geant4 has been reproduced in DD4hep
  - ➢ Crystal characteristics:
    - ✓ Dimension: 1.0 mm × 70.0 mm × 1.94 mm
    - ✓ Bending radius: 38.416 m
  - ➢ Gun:
    - ✓ 400 GeV protons
    - ✓ 100 events

Channeling works correctly in DD4hep

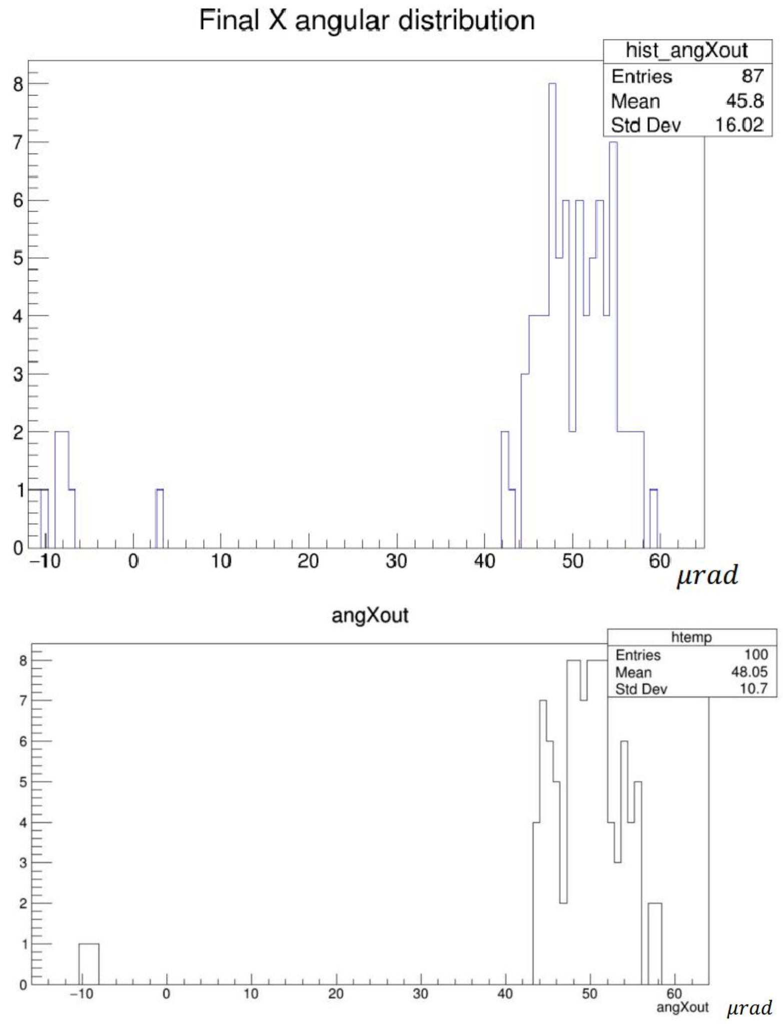


Fig. 5 The distribution of escaping angles. The upper picture is simulated by DD4hep and the lower is from Geant4

Channeling process

by Chiara Maccani

- Simulation of channeling still suffers a lot from the much slower speed than other processes
  - ➢ 141 minutes consumed when input 5 events generated by pythia
  - ➢ Caused mainly by the tracking of slow tracks

- Some solutions:
  - ➢ Add special cuts for particles with low energy
  - ➢ Using `G4StackingAction` to determined the simulation priority for each track

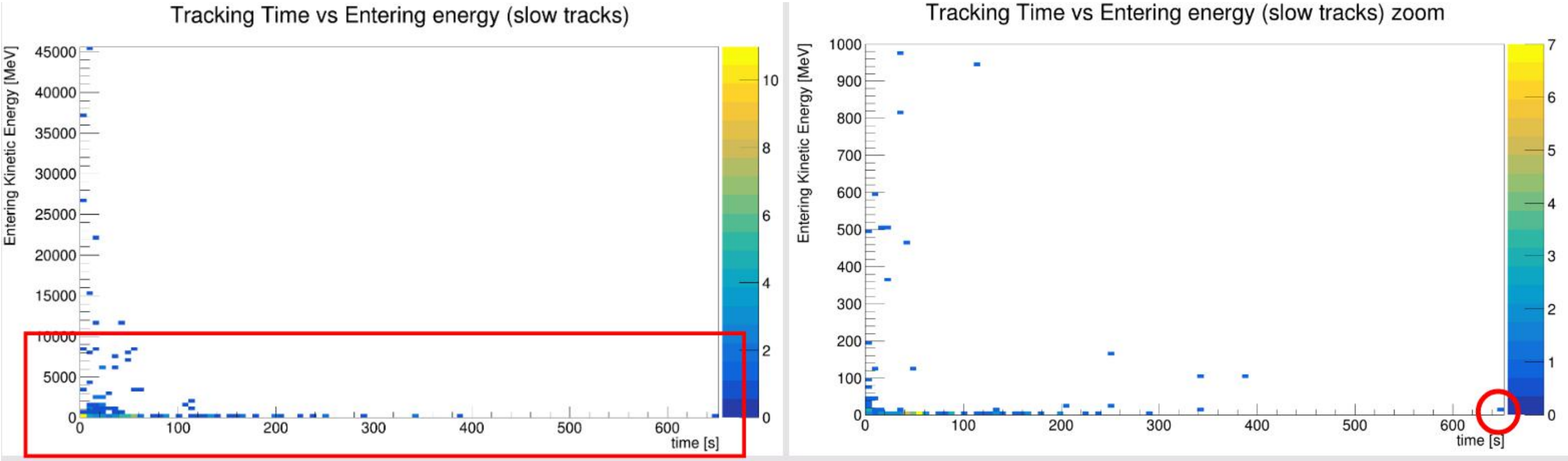


Fig. 6 Time of simualtion versus incident energy for different tracks

Three

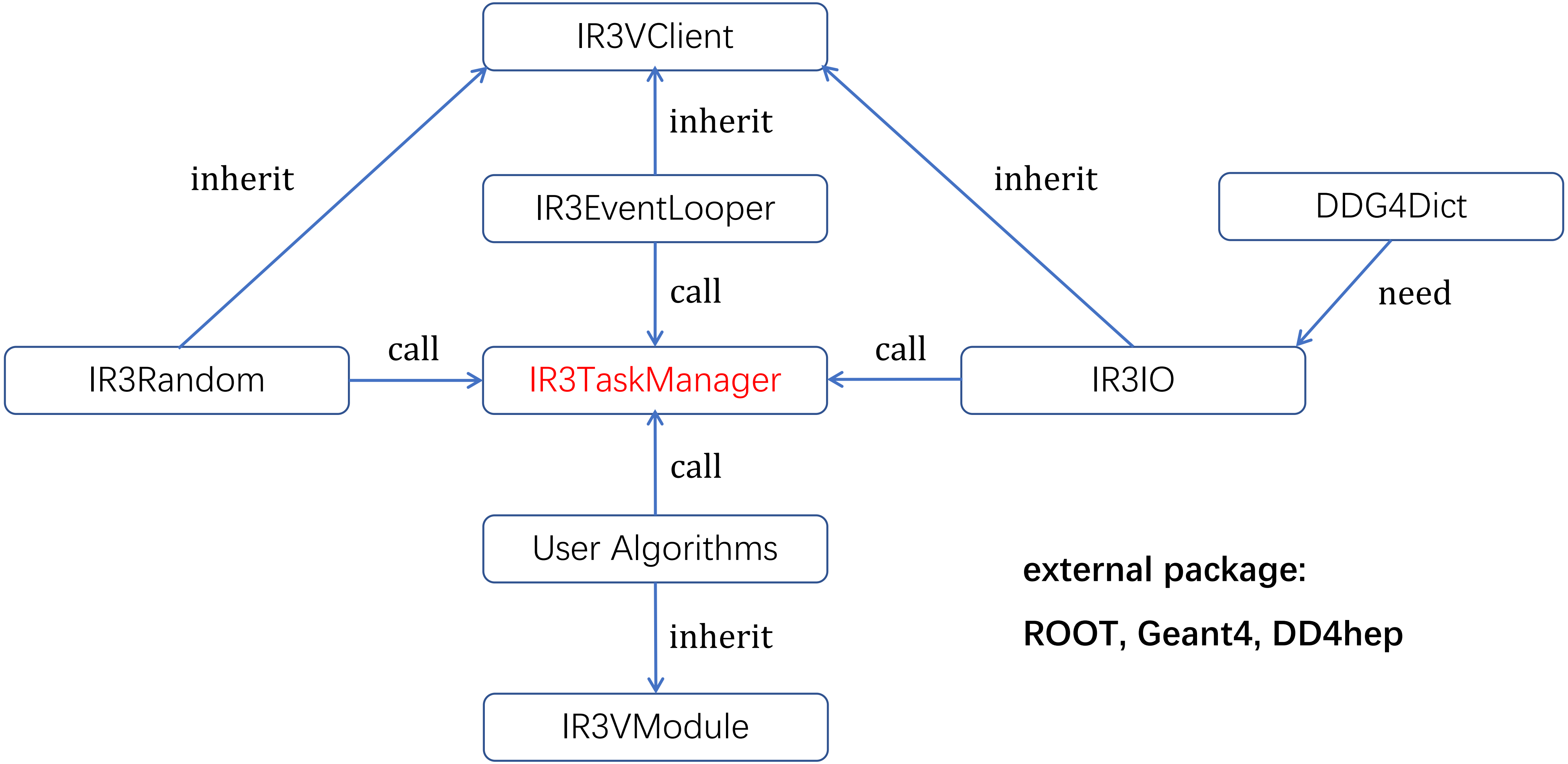
Analysis framework

tool to analyze the output file

Why an analysis framework

- Many procedures follow the simulation, including digitization, recontruction, calibration, analysis...
- A general, extendable and light-weighted analysis framework is necessary
- The event model is not a simple data structure which is handled by complex procedures
- The framework used in large experiment is complicated

Structure



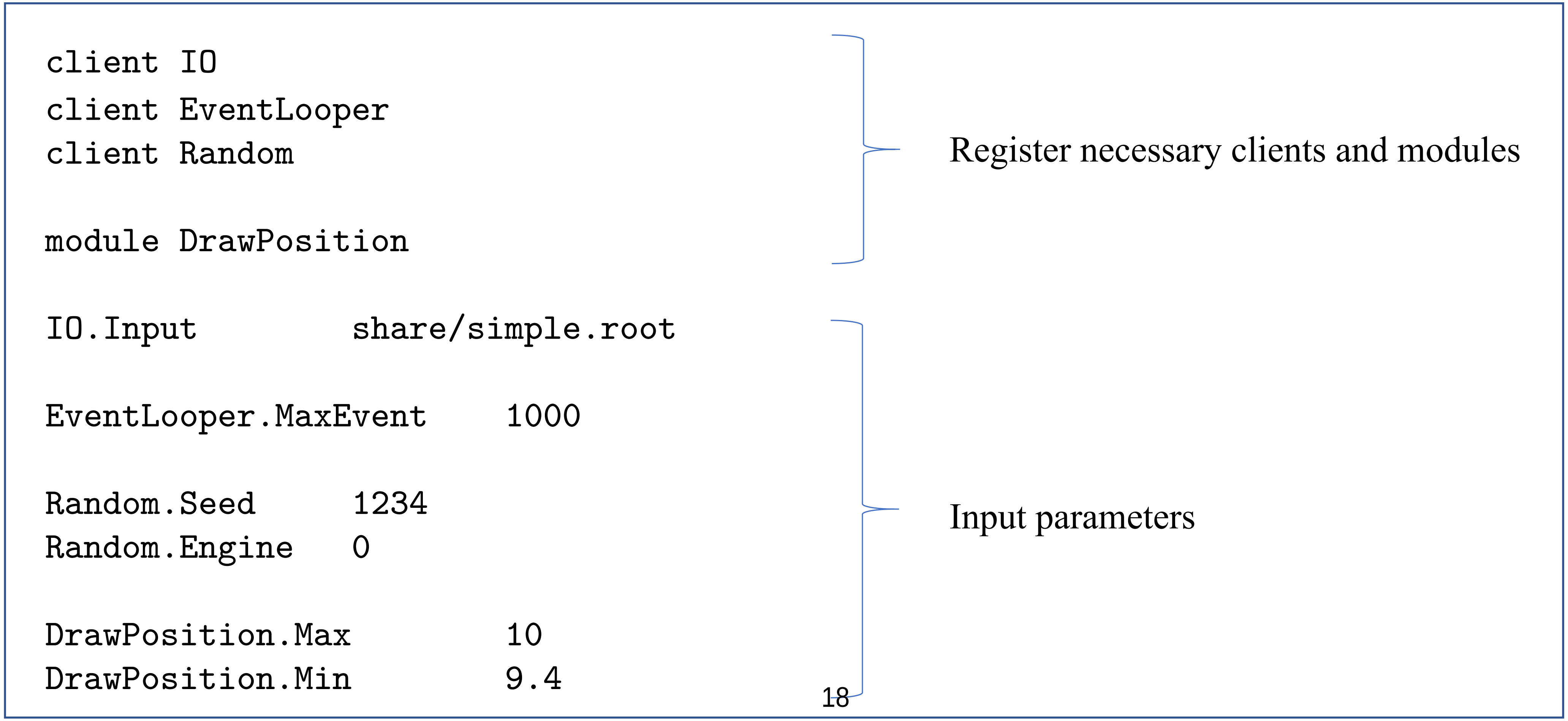
external package:

ROOT, Geant4, DD4hep

Fig. 7 Structure of IR3ana

Features

- Users are able to configure the job by a simple plain text file
- Convenient to implement the specilized algorithms or clients by users
- Provide interfaces for the following procedures including digitization, reconstruction and analysis



```
client IO
client EventLooper
client Random

module DrawPosition

IO.Input        share/simple.root

EventLooper.MaxEvent    1000

Random.Seed     1234
Random.Engine   0

DrawPosition.Max        10
DrawPosition.Min        9.4
```

Register necessary clients and modules

Input parameters

Application

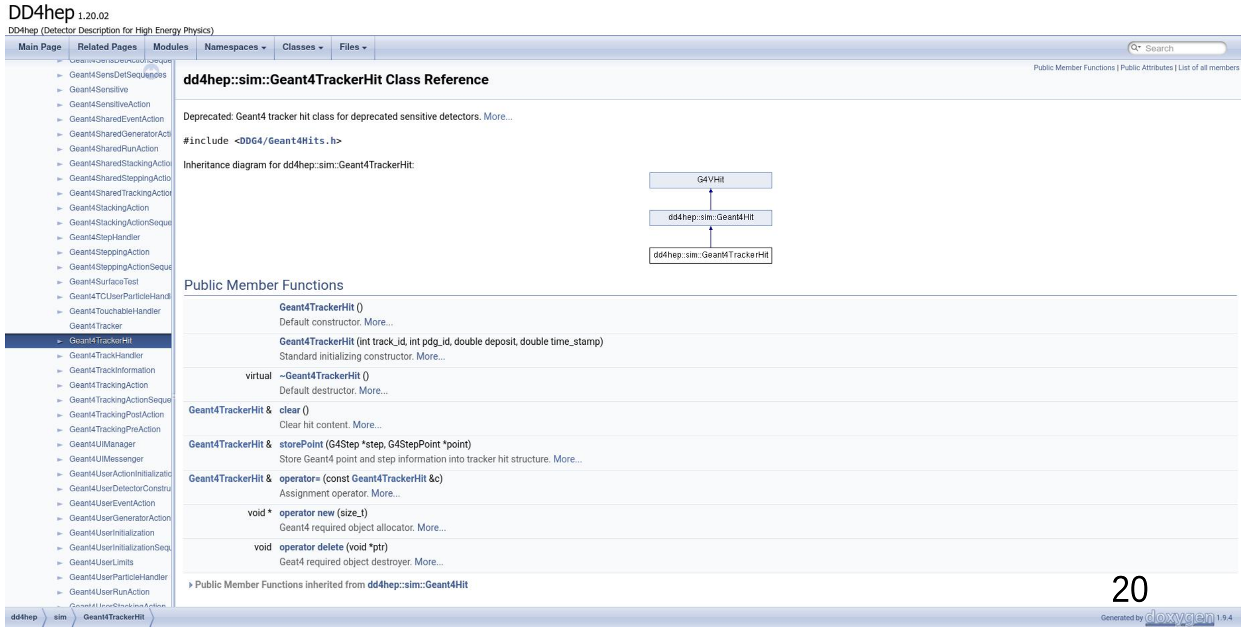
- IR3ana has already been used to deal with the output file of simualtion
- Occupancy of detectors has been obtained by Federico Zangari
- It can be applied for reconstruction in future

Doxygen

- Doxygen files of Geant4 and DD4hep with inline sources have been generated
  - ➢ Geant4 v11.0.1: https://gitlab.cern.ch/hmiao/geant4_doxygen
  - ➢ DD4hep v1.20.02: https://gitlab.cern.ch/hmiao/dd4hep_doxygen
- To run it (linux platform):

```
git clone ssh://git@gitlab.cern.ch:7999/hmiao/dd4hep_doxygen.git
cd dd4hep_doxygen/html
xdg-open classes.html
```

- The doxygen file will be running on a server of IHEP.



dd4hep::sim::Geant4TrackerHit Class Reference

Deprecated: Geant4 tracker hit class for deprecated sensitive detectors. More...

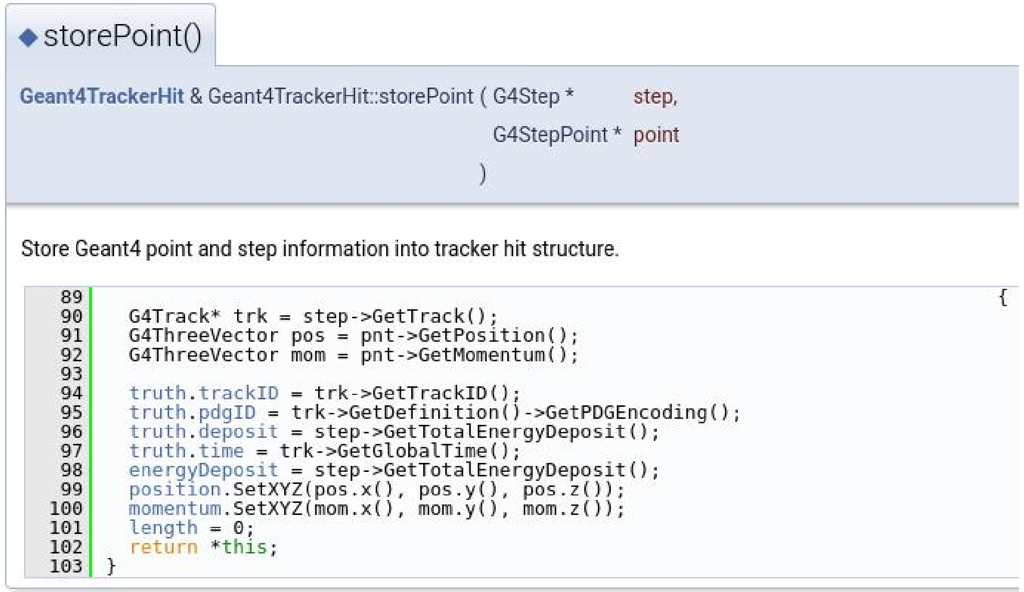
```
#include <DDG4/Geant4Hits.h>
```

Inheritance diagram for dd4hep::sim::Geant4TrackerHit:

Public Member Functions

	Geant4TrackerHit () Default constructor. More...
	Geant4TrackerHit (int track_id, int pdg_id, double deposit, double time_stamp) Standard initializing constructor. More...
virtual	~Geant4TrackerHit () Default destructor. More...
Geant4TrackerHit &	clear () Clear hit content. More...
Geant4TrackerHit &	storePoint (G4Step *step, G4StepPoint *point) Store Geant4 point and step information into tracker hit structure. More...
Geant4TrackerHit &	operator= (const Geant4TrackerHit &c) Assignment operator. More...
void *	operator new (size_t) Geant4 required object allocator. More...
void	operator delete (void *ptr) Geat4 required object destroyer. More...

▸ Public Member Functions inherited from dd4hep::sim::Geant4Hit



◆ storePoint()

```
Geant4TrackerHit & Geant4TrackerHit::storePoint ( G4Step *      step,
                                                  G4StepPoint * point
                                                )
```

Store Geant4 point and step information into tracker hit structure.

```
                                                                    {
  G4Track* trk = step->GetTrack();
  G4ThreeVector pos = pnt->GetPosition();
  G4ThreeVector mom = pnt->GetMomentum();

  truth.trackID = trk->GetTrackID();
  truth.pdgID = trk->GetDefinition()->GetPDGEncoding();
  truth.deposit = step->GetTotalEnergyDeposit();
  truth.time = trk->GetGlobalTime();
  energyDeposit = step->GetTotalEnergyDeposit();
  position.SetXYZ(pos.x(), pos.y(), pos.z());
  momentum.SetXYZ(mom.x(), mom.y(), mom.z());
  length = 0;
  return *this;
}
```

Four

Summary and next work

summary from software side and plan for future work

Summary and next work

- **Simulation framework:**
  - ✓ Simple geometry including essential facilities has been implemented
  - ✓ Provide two generators for users
  - ✓ Events can be visualized track by track
  - ➢ Implement General Particle Source (GPS) of Geant4
  - ➢ Further develop visualization module based on DDEve or JavaScript ROOT
  - ➢ Replace simple trackers with VeloPix
  - ➢ Add possible transition radiation detector
  - ➢ Compare physics lists provided by Geant4
  - ➢ Optimize the output event model

Summary and next work

- **Channeling process:**

✓ Channeling process provided by Geant4 runs correctly in DD4hep

➢ Implement channeling process to IR3 test simulation framework

➢ Find solutions to speed up the simulation

➢ Compare the results by G4Channeling with other models and existing measurement

- **Analysis framework:**

✓ Able to deal with the output file of simulation

✓ Occupancy and other informations have been obtained using IR3ana

➢ Continuously develop and maintain IR3ana

➢ More generic input&output interface for users

➢ Combine simulation and analysis to a unified framework

Thank you !