Simulation Framework in CEPCSW

Tao Lin

lintao@ihep.ac.cn
IHEP

The Joint Workshop of the CEPC Physics, Software and New Detector Concept
April 16, 2021
Yangzhou, China

Outline

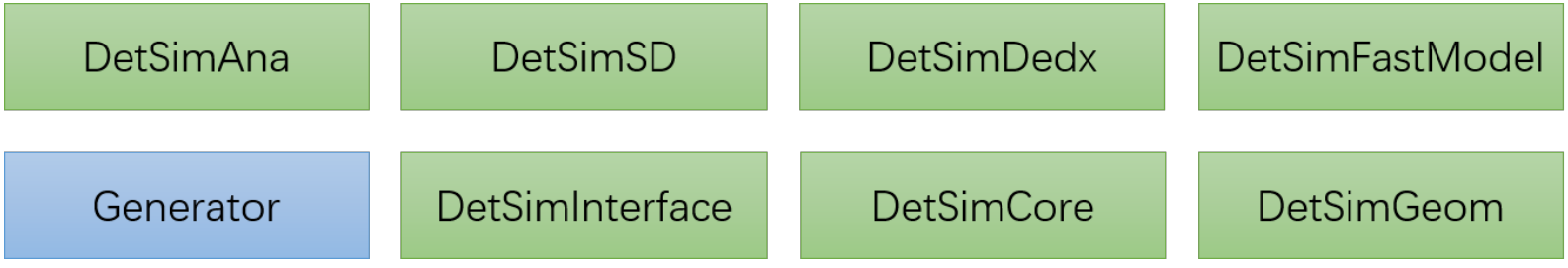
1. Introduction
2. Design and Implementation
3. Plans
4. Summary

Introduction

- Simulation plays an important role in
  - detector design and optimization
  - algorithms studies
- Instead of developing different standalone applications to simulate different detector options, a unified simulation framework has been developed in CEPCSW.
- Simulation framework provides the necessary software components to simplify the development of detector simulation with the unified interfaces.
  - Integration with the underlying framework
  - Event data model and MC truth
  - Detector descriptions
  - Detector responses
- The simulation software is implemented based on Key4hep project.

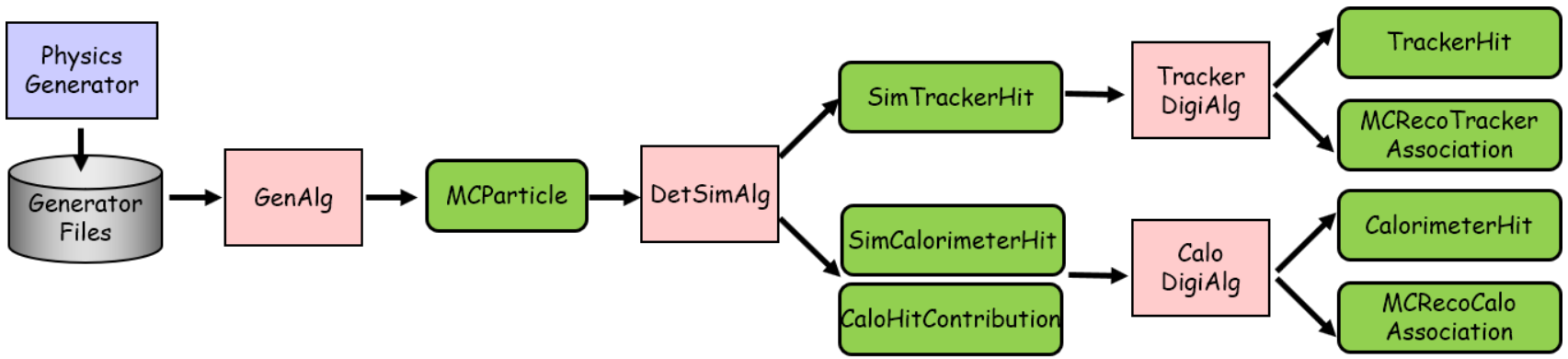
Design and Implementation

- A Geant4 based simulation framework has been developed based on Gaudi.
  - It supports the full and fast simulation transparently.
  - The default geometry is based on DD4hep detector model.
  - A Gaudi algorithm is in charge of the simulation workflow.
  - A customized G4 Run Manager breaks the original event loop in Geant4 and is integrated with the Gaudi algorithm.
  - Other services: random number, geometry service and so on.
- The core packages in CEPCSW:



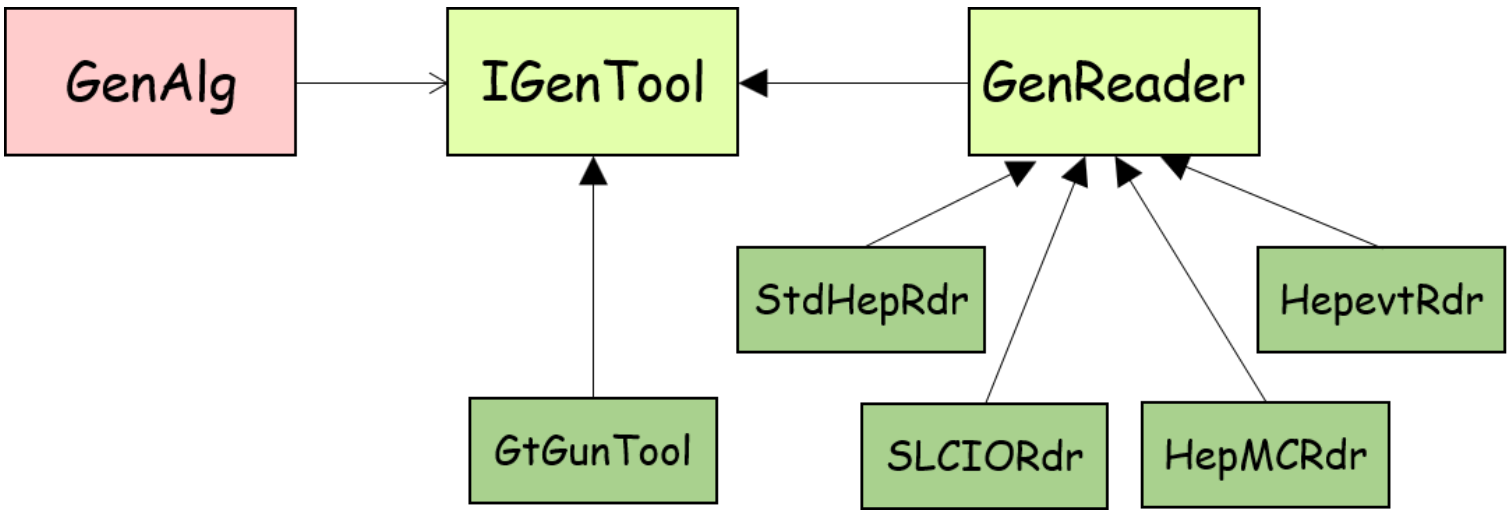
Simulation data flow with EDM4hep

- The simulation chain consists of:
  - physics generator,
  - detector simulation,
  - digitization.
- The EDM4hep is used as the input and output of each stage.



Physics generator interface

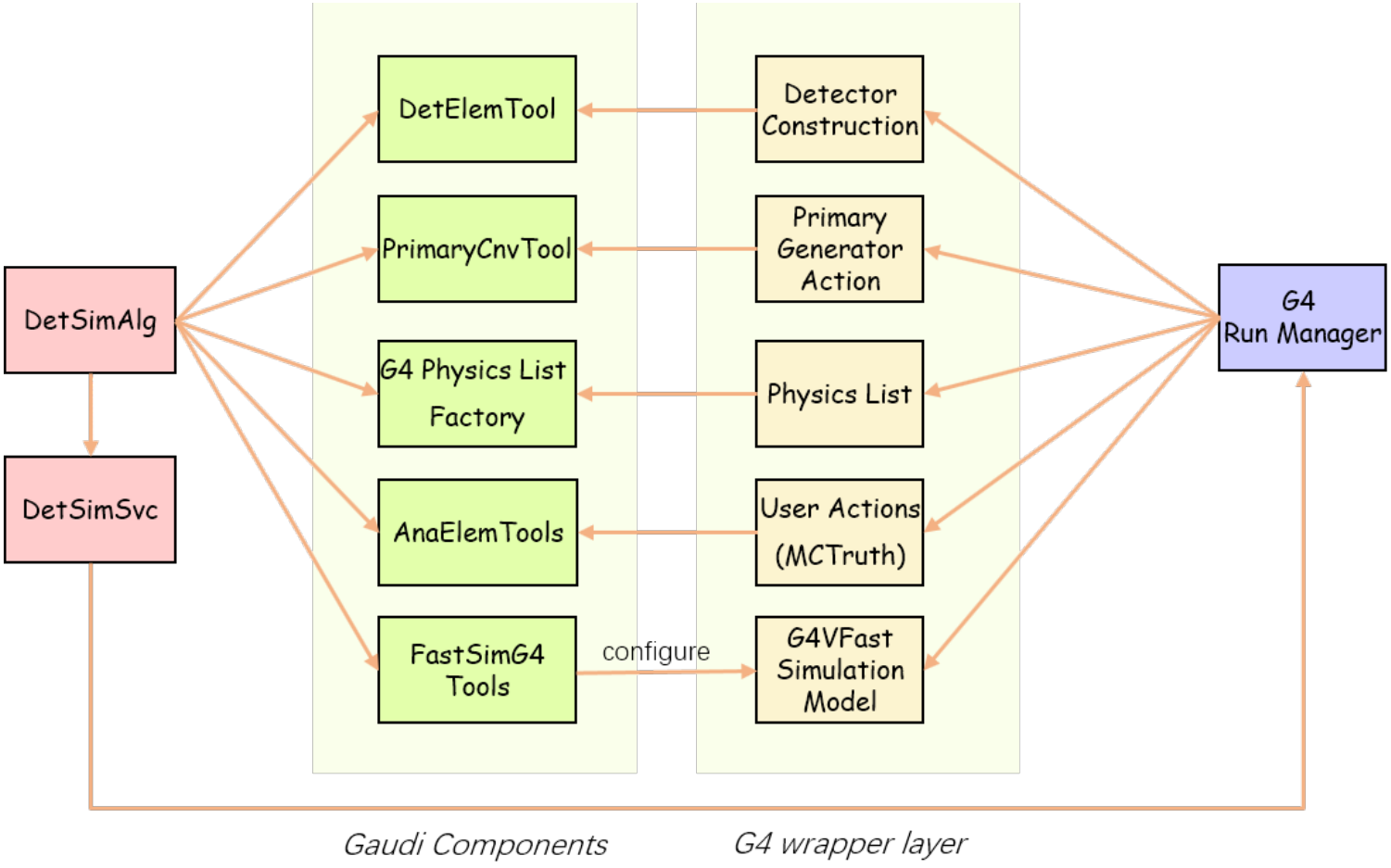
- The generator algorithm is configured with different tools
  - Particle gun,
  - Readers of different file formats



- Users can extend or develop new GenTools easily.

Integration with Geant4

- Full integration is done by developing a thin G4 wrapper layer:



Detector description and B-field with DD4hep

DD4hep is adopted to describe the geometry and B-field. Used features in the simulation:

- Compact files: the detector description parameters in XML file format.
- Detector Constructor: a sub detector implementation in C++, which could interpret the corresponding XML fragment.
- Readout: the sensitive part in the detector.
- Segmentation: a logical partition in the sensitive detector.
- ID Description: defines the Cell ID for each sensitive readout.

The package `DetSimGeom` is used to integrate the DD4hep and Geant4.

Physics processes

- Using the helper class `G4PhysListFactory` from Geant4 to create the physics list.
  - The default is `QGSP_BERT`.
  - More lists could be found in `https://geant4.kek.jp/lxr/source/physics_lists/lists/src/G4PhysListFactory.cc`
- Additional physics processes are also enabled
  - PAI or PAI photon model in EM processes
  - Step limiters
  - Fast simulation
- Optical processes are not enabled yet. But it is easy to enable.
- Details could be found in `https://github.com/cepc/CEPCSW/blob/master/Simulation/DetSimCore/src/DetSimAlg.cpp`

Detector responses

The MC hits are created in the detector response. Two types: Sim Tracker Hit and Sim Calorimeter Hit.

- The detector responses are implemented via the Geant4's sensitive detectors.
- Part of DDG4 is reused, such as the Geant4 hit objects and the existing SDs.
- For the dedicated detectors, following SDs are implemented:
  - Calorimeter: `CalorimeterSensDetTool`
  - Drift Chamber: `DriftChamberSensDetTool`
  - TPC: `TimeProjectionChamberSensDetTool`
- At the end of each event, the Geant4 hit objects are converted to EDM4hep objects.

MC truth

All the necessary relationships to re-build the relations between reconstructed particles and MC particles are stored:

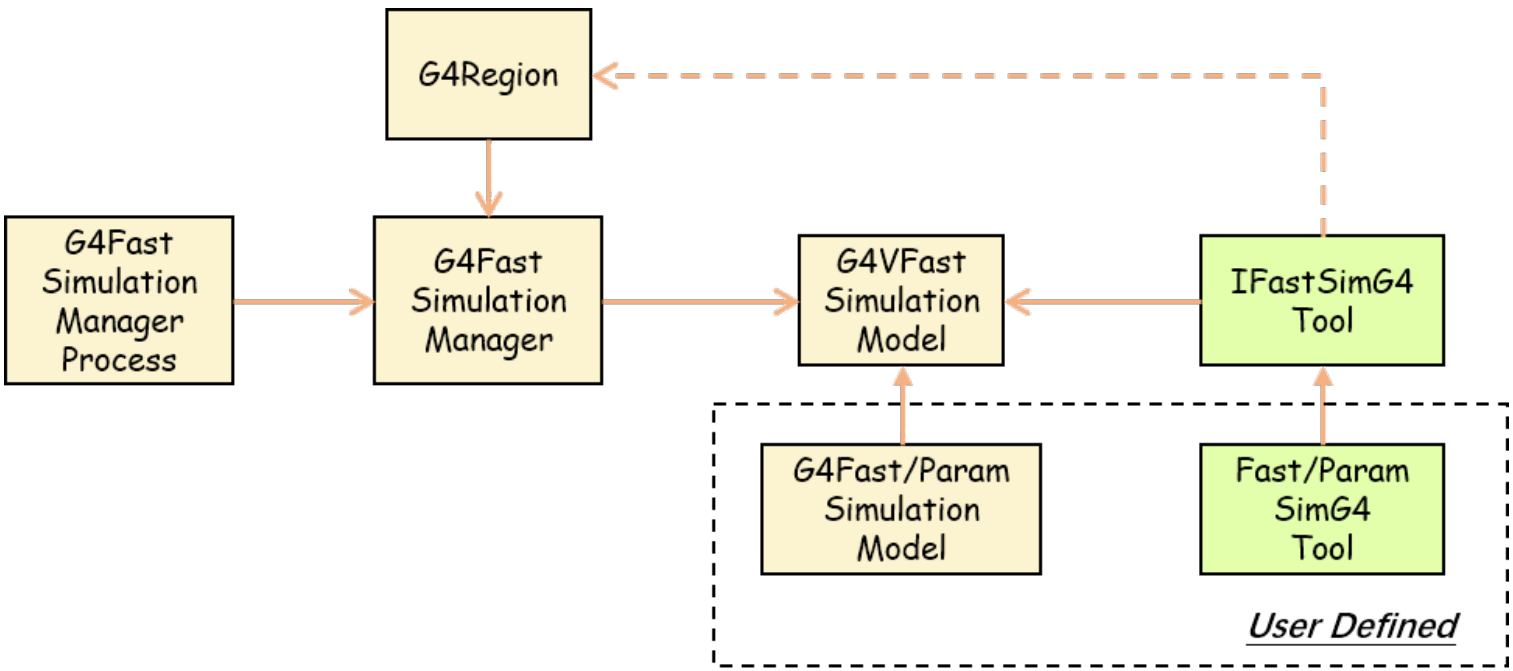
- The primary particles and decayed secondaries are stored in the MC particle collections.
- Given a hit object, the primary particle could be retrieved.
- There is a flag to indicate the hit is generated from secondary or not.

For the user defined MC truth information, users can maintain their own user trees by the ntuple service of Gaudi.

- Users are encouraged to contribute the implementation into the official repository.
- All the user output of the defined MC truth can be controlled in job options.

Fast simulation

- The fast simulation interface is integrated in Geant4.
- When a particle enters a region with fast simulation, Geant4 will trigger the fast simulation model.

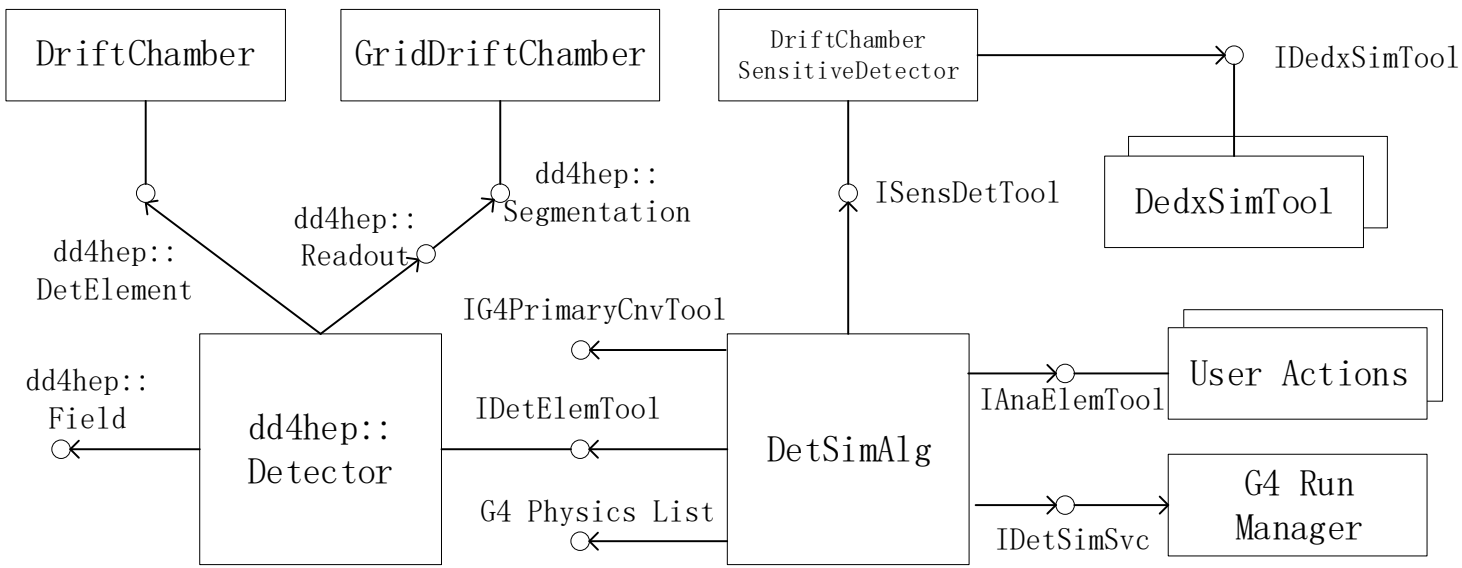


User interface

- All the simulation related components, which are implemented in Gaudi, could be configured in Python script.
  - Random seed numbers
  - GeomSvc and the detector description
  - Physics generator: particle gun or different readers
  - Geant4 batch macros or visualization macros
  - Physics processes
  - User actions
- The geometry could be customized in the XML files.
- The production cuts and step limiters could be tuned in XML via DD4hep.

An example: Drift Chamber

- The simulation of drift chamber is fully implemented in this framework.
  - The geometry is implemented with DD4hep.
  - The dE/dx or dN/dx is implemented in Gaudi tools.



Plans

Realistic simulation, such as non-uniformity of magnetic field, noise and background mixing etc.

- Develop a customized B-field class using DD4hep.
- Develop a set of tools to add noise at MC hit level.

Integration with various types of fast simulation tools.

- k4SimDelphes wrapper (`https://github.com/key4hep/k4SimDelphes`)
- fast ECAL simulation tools

Multi-threaded simulation prototype of dE/dx or dN/dx using GaudiHive.

- The simulation of avalanches of the primary ionization are run in different threads.

Summary

- The simulation framework has been developed and already used by several studies.
- New features are required and will be developed.

CEPCSW GitHub: `https://github.com/cepc/CEPCSW`

Thank you