Fluka2edepsim package

Lea Di Noto, Federico Ferraro,
Zahra Ghorbani Moghaddam,
Matteo Tenti, Paola Sala, Antonio Surdo

20-01-2021

(*Code Repository:* https://baltig.infn.it/dune/sand-fluka)

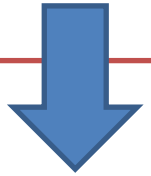
[not yet really moved to GitHub]

Present situation and goals

- SIMULATION codes: **-> FLUKA**
  -> GENIE+GEANT4

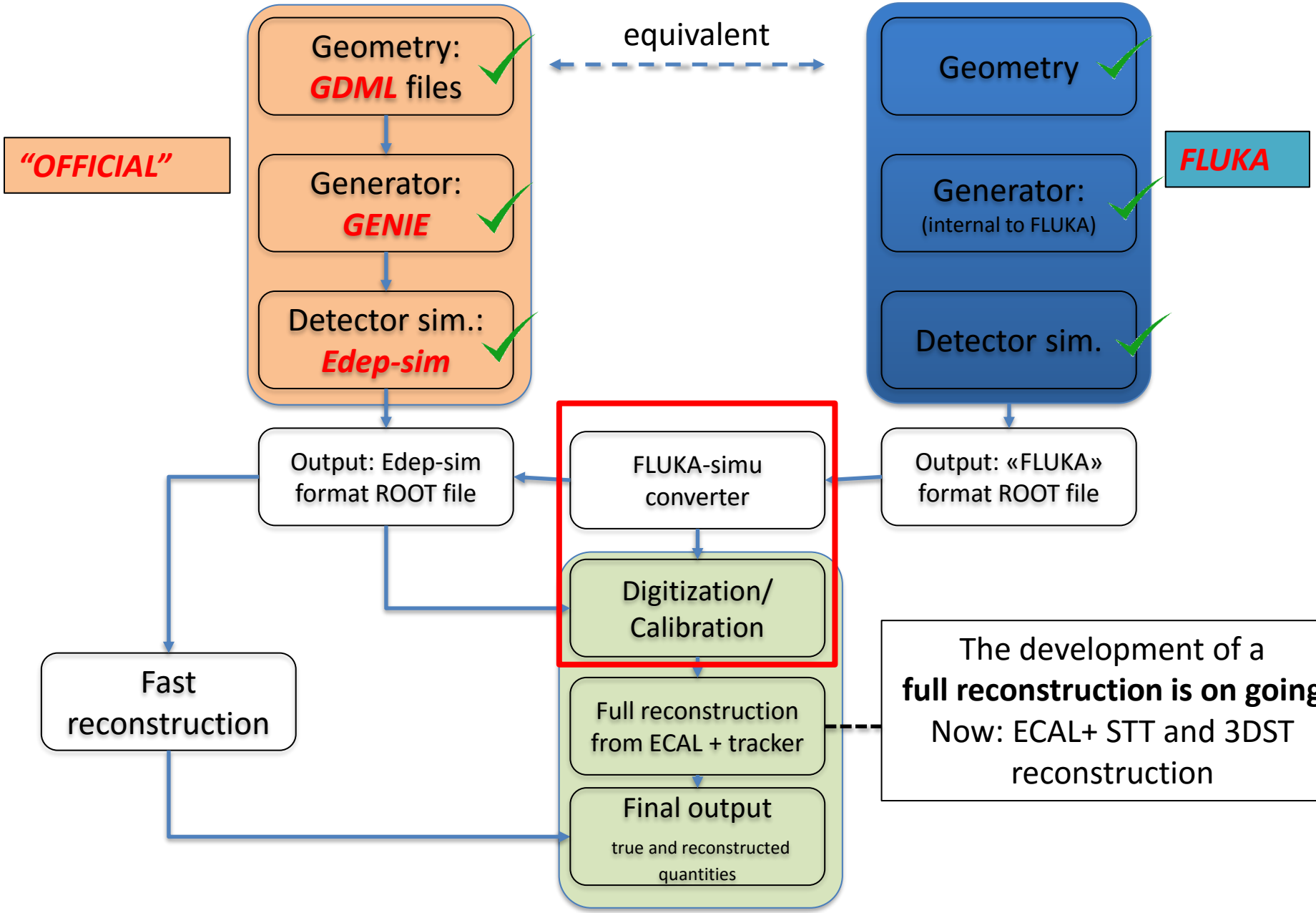
- *SAND* detector configurations:

ECal + STT (+ Lar target)	Fluka and Geant4
ECal + 3DST + STT (+ Lar target)	Fluka and Geant4
ECal + 3DST + TPC (+ Lar target)	only Geant4



Same output format from Simulation Codes as a prerequisite for a unique **General Analysis Framework**

The software in a nutshell



1st step: converting FLUKA output to Edep-sim

➢ FLUKA output

– **HeaderTree**

(Interaction and Vertex info)

– **HitsTree**

(particles entering volumes)

– **SttTree**

(Hits in STT and ECal)

– **CellTree**

(Hits in 3DST)

➢ EDEP-SIM format

- EDepSimEvents:
  - **TG4PrimaryVertex**
  - **TG4Trajectories**
  - **TG4HitSegment**
- Geometry info
- Input file + kinem

1st step: converting FLUKA output to Edep-sim

➢ FLUKA output

- **HeaderTree**
  (Interaction and vertex info)
- **HitsTree**
  (particles entering volumes)
- **SttTree**
  (Hits in STT and ECal)
- **CellTree**
  (Hits in 3DST)

NOT PRESENT IN FLUKA OUTPUT!!
We had to overcome to this!!

➢ EDEP-SIM format

- **EDepSimEvents**:
  - **TG4PrimaryVertex**
  - **TG4Trajectories**
  - **TG4HitSegment**
- Geometry info
- Input file + kinem

EDepSimEvents Tree structure

- ***TG4Event***

  One entry for each event

  - *RunId* [int] OK
  - *EventId* [int] OK

  > *Primaries* **[vector<TG4PrimaryVertex>]**

  One or more primary vertices

  > *Trajectories* **[vector<TG4Trajectory>]**

  Trajectories of all (primary and secondary) particles

  > *SegmentDetectors* **[map<string, vector<TG4HitSegment>>]**

  Map between each sensitive detector and the corresponding hit list

EDepSimEvents Tree structure

❑ *TG4PrimaryVertex*

- *Position* [TLorentzVector] OK

Remark: different Reference Systems in the simulation codes
FLUKA → origin in the SAND center
Edep-sim → origin in the beginning of Hall C

Coordinate translation (FLUKA → Edep-sim) : ***X → X***
Y → Y - 2384.73 mm
Z → Z + 23910.00 mm

- *GeneratorName* [string] OK
- *Reaction* [string] OK
- *Filename* [string] Empty for Fluka
- *InteractionNumber* [int] Empty for Fluka

  Index (or identifier) of the interaction in the kinematics file

EDepSimEvents Tree structure

❑ *TG4PrimaryVertex (contd)*

- *CrossSection* [float] Empty
  The cross section for the reaction that created this vertex
- *DiffCrossSection* [float] Empty
  The differential cross section for the kinematics of the reaction that created this vertex
- *Weight* [float] =1 OK
  The weight of the interaction. This will be set to one if the interaction is not reweighted. If the vertex is oversampled, this will be less than one.
- *Probability* [float] =1 OK
  The overall probability of the interaction that created this vertex. This includes the effect of the cross section, path length through the material, etc. This should be one if it is not filled

EDepSimEvents Tree structure

❑ *TG4PrimaryParticle*

- *TrackId* [int] OK
- *Name* [string] OK
- *PDGCode* [int] OK
- *Momentum* [TLorentzVector] OK

The initial momentum of the particle

❑ *TG4Trajectory*

- *TrackId* [int] OK
- *ParentId* [int] OK

TrackId of the most direct parent!

- *Name* [string] OK
- *PDGCode* [int] OK
- *InitialMomentum* [TLorentzVector] OK

> ***Points*** **[vector<TG4TrajectoryPoint>]**

EDepSimEvents Tree structure

❑ *TG4TrajectoryPoint*

- *Position* [TLorentzVector] OK
- *Momentum* [TVector3] OK
- *Process* [int] Fluka code
  NotDefined, Transportation, Electromagetic, Optical, Hadronic, PhotoLeptonHadron, Decay, General, Parameterization, UserDefined
- *Subprocess* [int] Fluka code
  EMCoulombScattering, EMIonization, EMBremsstrahlung, EMPairProdByCharged, EMNuclearStopping, EMMultipleScattering, EMPhotoelectric, EMComptonScattering, EMGammaConversion, HadronElastic, HadronInelastic, HadronCapture, HadronChargeExchange, GeneralStepLimit

EDepSimEvents Tree structure *(contd)*

❑ *TG4HitSegment*

- *TrackLength* [float] Empty (not used)
  The total charged track length in this hit. This includes the contribution from all of the secondary particles (e.g. delta-rays) that are included in this hit.
- *Start* [TLorentzVector] OK
- *Stop* [TLorentzVector] OK
- *EnergyDeposit* [float] OK
  The total energy deposit in this hit
- *SecondaryDeposit* [float] = 0 OK
  The "secondary" energy deposit in this hit. Generally, this is used to help simulate the amount of energy emitted as scintillation light, i.e. optical photons, and is part of the total energy deposit. The remaining energy will be deposited as ionization.

EDepSimEvents Tree structure

❑ *TG4HitSegment*

- *Contrib* [vector<int>] OK
  The TrackId for each trajectory that contributed to this hit

NEW RULE IMPOSED IN FLUKA SIMU:
each particle present in TG4HitSegment (in contrib) must be present in the TG4Trajectory

- *PrimaryId* [int] ~ OK
  The track id of the "**most important primary particle"** associated with this hit segment:
  - particle generated in the main interaction vertex
  - particle generated in a secondary vertex as a decay product

We have taken into account only of :
-pi0 decay (gamma's daughter have gamma's TrackId as PrimaryId)
-muon decay
In edep-sim other process are probably considered.

2nd step: making Digitization code compatible with FLUKA simulations

Why Digitization cannot be the same for both:

✓ Digitization needs detector Geometry (TGeoManager)

The code by itself recognize if the input file comes from FLUKA or GEANT4 and will load the proper Geometry infos

✓ The physical simulation of ECAL is different
→so the tuned parameters must be different

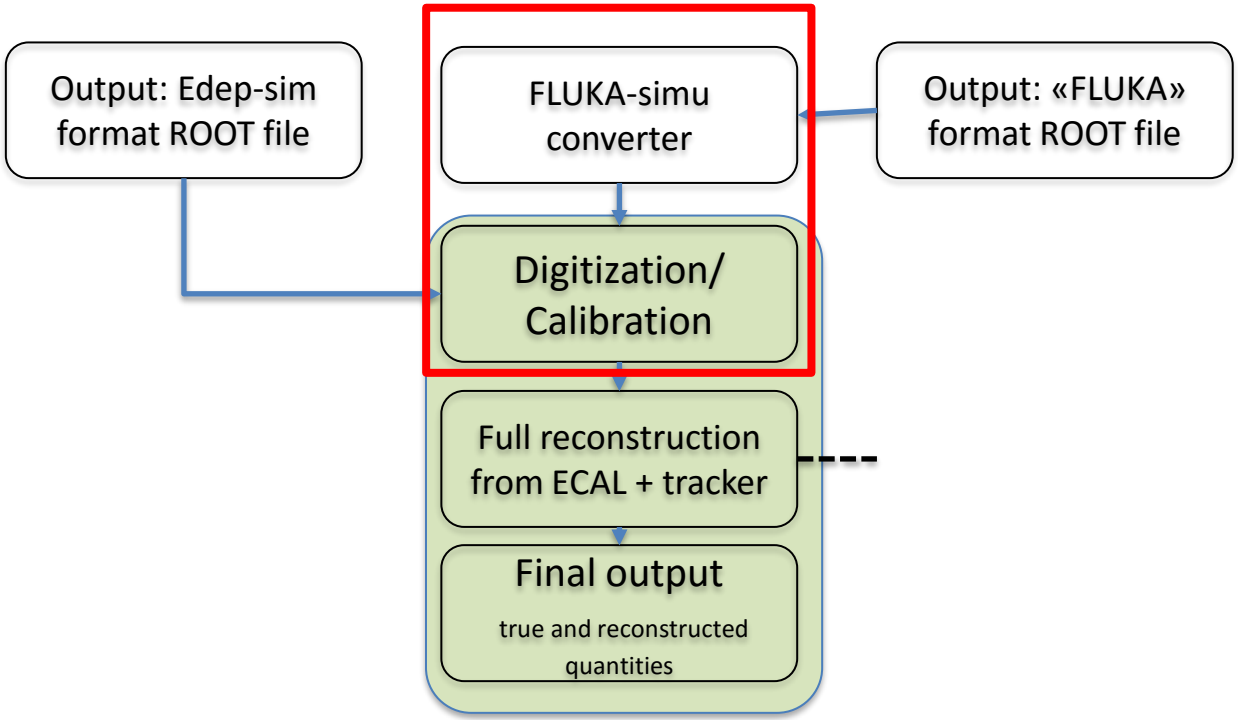
so *Digitization* will run differently:

- STT digit (same function for both)
- ECAL digit (different function due to different tunings)

scintillating layers in Geant4 while scintillating fibers in Fluka

Dedicated simulations (muons, electrons towards ecal module) are being analyzed for checking the equivalent output

In conclusion



Reco and *Analysis* codes can work on both FLUKA and GEANT4 outputs!!
BUT
special attention should be put for using PrimaryId variable