Truth Associations Discucssion

Association Logic | Main vs. Proposed

Main: current logic applied in CalorimeterClusterRecoCoG

For each cluster **do:**

1. **Find** cell with largest energy deposit
2. Get matching *sim hit* based on cell ID
3. Grab first *contribution* to sim hit and set corresponding particle as association
   - ☞ with weight = 1.0

Reminders:

- "sim hit" = sum of G4Hits in a cell
- "contribution" = G4Hit

Proposed Change: thoughts?

For each cluster **do:**

1. **For each** cell **in** cluster **do:**
   a) Find matching sim hit based on cell ID
   b) **For each** contributions **to** sim hit **do:**
      a) Get corresponding particle
      b) Walk back through parents to find first *primary*
      c) Increment contributed sum for that primary
2. **Create** an association for each contributing primary
   - ☞ with weight = contributed energy / total energy

Notes:

- "primary" = particle w/ generator status == 1
- See discussions in Issue#1475, PR#1396, and June 4th Reco Meeting

Contributions | How to Work With Them?

- **Contributions:** (i.e. G4Hits) *not* available in simulation campaign output
  - BUT will be stored by default if you run EICrecon locally
  - Could be stored for specialized productions?

- **Right:** how to access contributions to a given cluster
  - Example macro available in snippets repo here
  - Uses PODIO Reader, but pure ROOT logic will be similar

```
// loop over clusters
for (size_t iClust = 0; edm4eic::Cluster cluster : clusters) {

  // loop through reconstructed cluster hits (cells)
  for (size_t iRec = 0; edm4eic::CalorimeterHit rec : cluster.getHits()) {

    // grab cell ID
    const uint64_t recCellID = rec.getCellID();

    // loop over sim hits
    for (size_t iSim = 0; edm4hep::SimCalorimeterHit sim : sim_hits) {

      // match sim-to-reco hit based on cell ID
      const uint64_t simCellID  = sim.getCellID();
      const bool     isSameCell = (recCellID == simCellID);
      if (!isSameCell) continue;

      // now loop over contributions
      for (size_t iContrib = 0; edm4hep::CaloHitContribution contrib : sim.getContributions()) {

        // grab corresponding particle
        auto particle = contrib.getParticle();
```

MCParticles | What's Stored?

- **Note:** by default, only particles created inside **tracking volume** are stored in MCParticles
  - So associations *always* point back to particles in tracking volume
  - All G4Hits in a calo are assigned particle that exits tracking volume

- **However:** "detailed shower mode" should store more particles
  - Not turned on by default in npsim [c.f. this issue]
    - ☞ Can be turned on for standalone studies/specialized productions
  - **Long term:** might be useful to have a mode in between normal and detailed
    - › e.g. could add secondaries from calos which have an end vertex outside given calo
    - › Would need to implement *in DD4hep*

Tips n’ Tricks

Tips n' Tricks | Turn on Detailed Shower Mode

```
npsim --enableDetailedShowerMode --steeringFile $steerer --compactFile $compact -I $input -N $numevts --outputFile $output
```

- This option turns on detailed shower mode
- A few extra options that might be useful for looking at showers in detail…

```
from DDSim.DD4hepSimulation import DD4hepSimulation
from g4units import mm, GeV, MeV, degree

SIM = DD4hepSimulation()
SIM.part.keepAllParticles = True
SIM.part.minimalKineticEnergy = 0*MeV
SIM.filter.calo = "edep0"
```

Tips n’ Tricks | Adjust Tracking Volume

- Can adjust which detectors are included in tracking volume in compact file with this option

```
<detectors>
  <detector
    id="BarrelTOF_ID"
    name="BarrelTOF"
    type="epic_TOFBarrel"
    readout="TOFBarrelHits"
    insideTrackingVolume="true">
```

Tips n' Tricks | Generator Status

Generator Status	Value
Empty	0 (1<<0)
Stable	1 (1<<1)
Decayed	2 (1<<2)
Documentation(?)	3 (1<<3)
Beam Particle	4 (1<<4)
Other	9 (1<<9)

- See here for definitions, usage

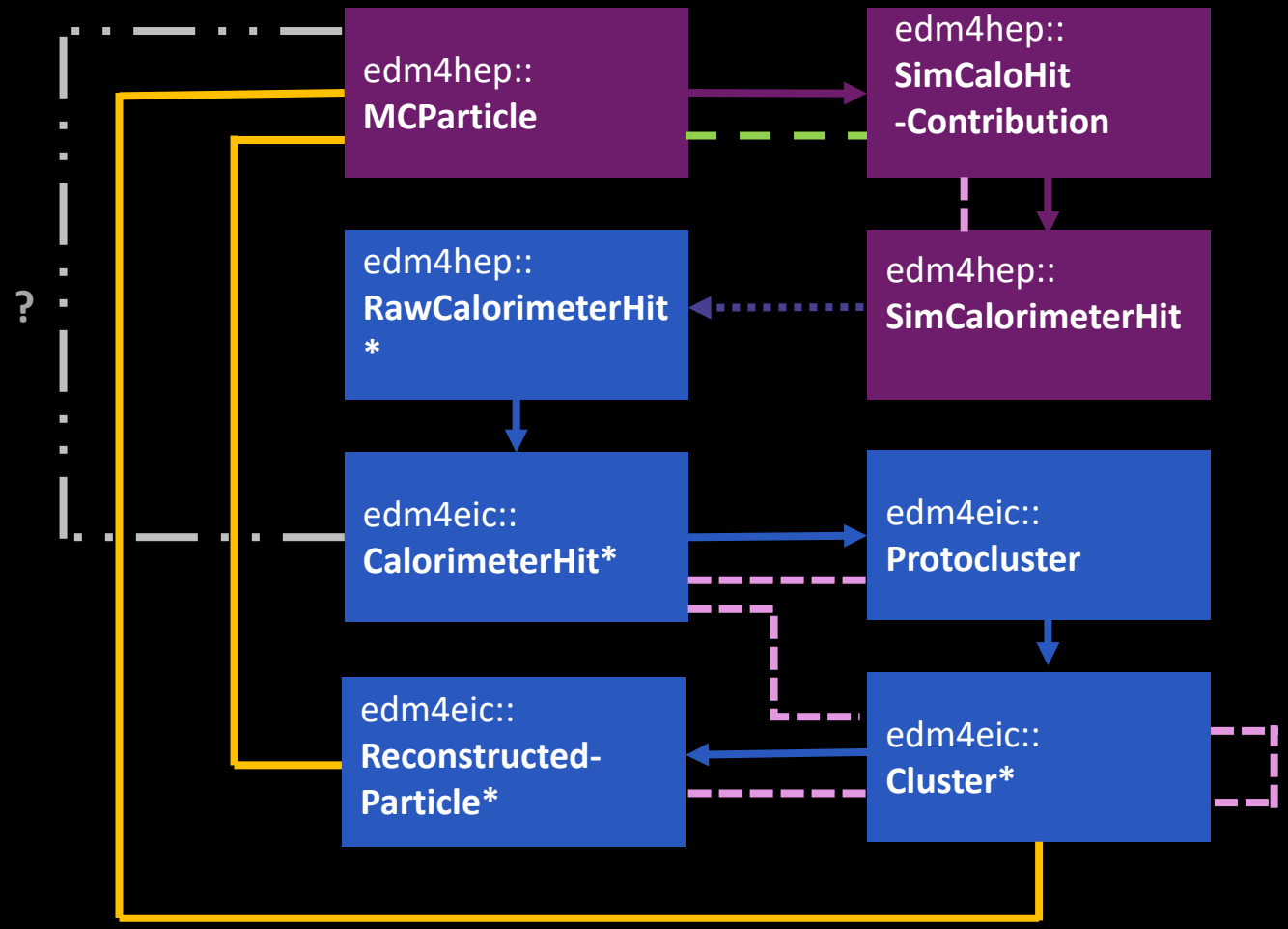
Tips n' Tricks | Simulation Status

Simulation Status	Accessor	Value
Created by simulation	isCreatedInSimulation()	1024 (1<<10)
Backscatter from calo	isBackscatter()	2048 (1<<11)
Interacted in a calo region	isDecayedInCalorimeter()	4096 (1<<12)
Interacted in a tracking region	isDecayedInTracker()	8192 (1<<13)
Stopped by simulation	isStopped()	16384 (1<<14)
Left world volume undecayed	hasLeftDetector()	32768 (1<<15)
Particle's vertex is not parent endpoint	vertexIsNotEndpointOfParent()	65536 (1<<16)
Has been "overlayed" by simulation	isOverlay()	131072 (1<<17)

- See here for status definitions
- And here for accessor definitions

Backup

Backup | Datatypes & Their Connections



Legend:

- = Digitization
- = Assocation
- = One-to-one relation
- = One-to-many relation
- = DD4hep
- = EICrecon

* Saved to EICrecon output by default

Backup | edm4hep::SimCaloHitContribution


```
#------------- CaloHitContribution
edm4hep::CaloHitContribution:
  Description: "Monte Carlo contribution to SimCalorimeterHit"
  Author: "F.Gaede, DESY"
  Members:
    - int32_t   PDG                      //PDG code of the shower particle that caused this contribution.
    - float energy                    //energy in [GeV] of the this contribution
    - float time                      //time in [ns] of this contribution
    - edm4hep::Vector3f stepPosition   //position of this energy deposition (step) [mm]
  OneToOneRelations:
    - edm4hep::MCParticle particle     //primary MCParticle that caused the shower responsible for this contribution to the hit.
```

Backup | edm4hep::SimCalorimeterHit


```
#------------ SimCalorimeterHit
edm4hep::SimCalorimeterHit:
  Description: "Simulated calorimeter hit"
  Author: "F.Gaede, DESY"
  Members:
    - uint64_t cellID       //ID of the sensor that created this hit
    - float energy                    //energy of the hit in [GeV].
    - edm4hep::Vector3f position      //position of the hit in world coordinates in [mm].
  OneToManyRelations:
    - edm4hep::CaloHitContribution contributions  //Monte Carlo step contribution - parallel to particle
```

Backup | edm4hep::RawCalorimeterHit

```
#-------------  RawCalorimeterHit
edm4hep::RawCalorimeterHit:
  Description: "Raw calorimeter hit"
  Author: "F.Gaede, DESY"
  Members:
    - uint64_t cellID   //detector specific (geometrical) cell id.
    - int32_t amplitude              //amplitude of the hit in ADC counts.
    - int32_t timeStamp              //time stamp for the hit.
```

Backup | edm4eic::RawCalorimeterHit

```
edm4eic::RawCalorimeterHit:
  Description: "Raw (digitized) calorimeter hit"
  Author: "W. Armstrong, S. Joosten"
  Members:
    - uint64_t          cellID          // The detector specific (geometrical) cell id.
    - uint64_t          amplitude       // The magnitude of the hit in ADC counts.
      ## @TODO: should we also add integral and time-over-threshold (ToT) here? Or should
      ##        those all be different raw sensor types? Amplitude is
      ##        really not what most calorimetry sensors will give us AFAIK...
    - uint64_t          timeStamp       // Timing in TDC
```

Backup | edm4eic::CalorimeterHit

```
edm4eic::CalorimeterHit:
  Description: "Calorimeter hit"
  Author: "W. Armstrong, S. Joosten"
  Members:
    - uint64_t          cellID          // The detector specific (geometrical) cell id.
    - float             energy          // The energy for this hit in [GeV].
    - float             energyError     // Error on energy [GeV].
    - float             time            // The time of the hit in [ns].
    - float             timeError       // Error on the time
    - edm4hep::Vector3f position        // The global position of the hit in world coordinates [mm].
    - edm4hep::Vector3f dimension       // The dimension information of the cell [mm].
    - int32_t           sector          // Sector that this hit occurred in
    - int32_t           layer           // Layer that the hit occurred in
    - edm4hep::Vector3f local           // The local coordinates of the hit in the detector segment [mm].
```

Backup | edm4eic::Protocluster

```
edm4eic::ProtoCluster:
  Description: "Collection of hits identified by the clustering algorithm to belong together"
  Author: "S. Joosten"
  OneToManyRelations:
    - edm4eic::CalorimeterHit hits       // Hits associated with this cluster
  VectorMembers:
    - float                   weights    // Weight for each of the hits, mirrors hits array
```

Backup | edm4eic::Protocluster

```
edm4eic::ProtoCluster:
  Description: "Collection of hits identified by the clustering algorithm to belong together"
  Author: "S. Joosten"
  OneToManyRelations:
    - edm4eic::CalorimeterHit hits       // Hits associated with this cluster
  VectorMembers:
    - float                   weights    // Weight for each of the hits, mirrors hits array
```

Backup | edm4eic::Cluster

```
edm4eic::Cluster:
  Description: "EIC hit cluster, reworked to more closely resemble EDM4hep"
  Author: "W. Armstrong, S. Joosten, C.Peng"
  Members:
    # main variables
    - int32_t          type              // Flag-word that defines the type of the cluster
    - float            energy            // Reconstructed energy of the cluster [GeV].
    - float            energyError       // Error on the cluster energy [GeV]
    - float            time              // [ns]
    - float            timeError         // Error on the cluster time
    - uint32_t         nhits             // Number of hits in the cluster.
    - edm4hep::Vector3f position          // Global position of the cluster [mm].
    - edm4eic::Cov3f   positionError     // Covariance matrix of the position (6 Parameters).
    - float            intrinsicTheta    // Intrinsic cluster propagation direction polar angle [rad]
    - float            intrinsicPhi      // Intrinsic cluster propagation direction azimuthal angle [rad]
    - edm4eic::Cov2f   intrinsicDirectionError // Error on the intrinsic cluster propagation direction
  VectorMembers:
    - float            shapeParameters   // Should be set in metadata, for now it's a list of -- radius [mm], dispersion [mm], 2 entries for
    - float            hitContributions  // Energy contributions of the hits. Runs parallel to ::hits()
    - float            subdetectorEnergies // Energies observed in each subdetector used for this cluster.
  OneToManyRelations:
    - edm4eic::Cluster        clusters    // Clusters that have been combined to form this cluster
    - edm4eic::CalorimeterHit hits        // Hits that have been combined to form this cluster
    - edm4hep::ParticleID     particleIDs // Particle IDs sorted by likelihood
```

Backup | edm4eic::ReconstructedParticle

```
edm4eic::ReconstructedParticle:
  Description: "EIC Reconstructed Particle"
  Author: "W. Armstrong, S. Joosten, F. Gaede"
  Members:
    - int32_t           type              // type of reconstructed particle. Check/set collection parameters ReconstructedParticleTypeNames and
    - float             energy            // [GeV] energy of the reconstructed particle. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f momentum          // [GeV] particle momentum. Four momentum state is not kept consistent internally.
    - edm4hep::Vector3f referencePoint    // [mm] reference, i.e. where the particle has been measured
    - float             charge            // charge of the reconstructed particle.
    - float             mass              // [GeV] mass of the reconstructed particle, set independently from four vector. Four momentum state
    - float             goodnessOfPID     // overall goodness of the PID on a scale of [0;1]
    - edm4eic::Cov4f    covMatrix         // covariance matrix of the reconstructed particle 4vector (10 parameters).
    ##@TODO: deviation from EDM4hep: store explicit PDG ID here. Needs to be discussed how we
    ##       move forward as this could easiliy become unwieldy without this information here.
    ##       The only acceptable alternative would be to store reconstructed identified
    ##       particles in separate collections for the different particle types (which would
    ##       require some algorithmic changes but might work. Doing both might even make
    ##       sense. Needs some discussion, note that PID is more emphasized in NP than
    ##       HEP).
    - int32_t           PDG               // PDG code for this particle
    ## @TODO: Do we need timing info? Or do we rely on the start vertex time?
  OneToOneRelations:
    - edm4eic::Vertex      startVertex    // Start vertex associated to this particle
    - edm4hep::ParticleID  particleIDUsed // particle ID used for the kinematics of this particle
  OneToManyRelations:
    - edm4eic::Cluster     clusters       // Clusters used for this particle
    - edm4eic::Track       tracks         // Tracks used for this particle
    - edm4eic::ReconstructedParticle particles // Reconstructed particles that have been combined to this particle
    - edm4hep::ParticleID  particleIDs    // All associated particle IDs for this particle (not sorted by likelihood)
  ExtraCode:
    declaration: "
      bool isCompound() const {return particles_size() > 0;}\n
      "
```

Backup | edm4eic::MCRecoClusterParticleAssoc.

```
edm4eic::MCRecoClusterParticleAssociation:
  Description: "Association between a Cluster and a MCParticle"
  Author : "S. Joosten"
  Members:
    - uint32_t          simID           // Index of corresponding MCParticle (position in MCParticles array)
    - uint32_t          recID           // Index of corresponding Cluster (position in Clusters array)
    - float             weight          // weight of this association
  OneToOneRelations:
    - edm4eic::Cluster  rec             // reference to the cluster
    - edm4hep::MCParticle sim           // reference to the Monte-Carlo particle
```