

Data model changes for electron finder

- Tyler recently completed work on track projections
- Next step for electron finder (and other milestones) is track-cluster matching
- At a very minimum (as a final step) we need RecoPartice \leftrightarrow Reco cluster associations to be stored
- Currently we have a separate types for each association (next slide)
- Also options (changes needed) to store oneToMany relationships on objects (cluster, tracks, RecoParticles)

edm4eic::MCRecoParticleAssociation:

Description: "Used to keep track of the correspondence between MC and reconstructed particles"

Author: "S. Joosten"

Members:

- uint32_t simID // Index of corresponding MCParticle (position in MCParticles array)
- uint32_t recID // Index of corresponding ReconstructedParticle (position in ReconstructedParticles array)
- float weight // weight of this association

OneToOneRelations :

- edm4eic::ReconstructedParticle rec // reference to the reconstructed particle
- edm4hep::MCParticle sim // reference to the Monte-Carlo particle

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

Would be nice to have a generic association? But besides using uint32_t id1, id2 is this possible?

Current Proposal

```
495 +
496 +   edm4eic::RecoClusterParticleAssociation:
497 +   Description: "Association between a Cluster and a
      ReconstructedParticle"
498 +   Author : "D. Brandenburg"
499 +   Members:
500 +       - float          weight          // weight of this association
501 +   OneToOneRelations:
502 +       - edm4eic::Cluster  clu          // reference to the cluster
503 +       - edm4eic::ReconstructedParticle plc          // reference to the
      Reco particle
```



Assumes adoption of Dmitri's suggestion to drop uint32_t IDs

Specifically for RecoParticle & cluster

Still missing (skipping) track <-> cluster association

Should we add an (nearly) identical type for track & cluster?

RecoParticle Model

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 Reconst
- float           energy         // [GeV] energy of the reconstructed particle. Four momentum state is not
- edm4hep::Vector3f momentum     // [GeV] particle momentum. Four momentum state is not kept consistent int
- 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 v
- 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 easilly become unwiely 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 likelihooc
```

ExtraCode:

```
declaration: "
    bool isCompound() const {return particles_size() > 0;}
"
```

Cluster model

```
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
    - 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
```

Need oneToMany -> track,
RecoParticle?

Track Model

edm4eic::Track:

Description: "Track information at the vertex"

Author: "S. Joosten"

Members:

```
- int32_t          type          // Flag that defines the type of track
- float           chi2          // Total chi2 (sum) of the track fit
- int32_t          ndf          // Numbers of degrees of freedom of the track fit
- edm4hep::Vector3f momentum    // Track 3-momentum at the vertex [GeV]
- edm4eic::Cov3f   momentumError // Covariance matrix on the momentum
- float           time          // Track time at the vertex [ns]
- float           timeError     // Error on the track vertex time
- float           charge        // Particle charge
```

OneToOneRelations:

```
- edm4eic::Trajectory trajectory // Trajectory of this track
- edm4eic::Vertex   vertex       // Track vertex of this track
```

OneToManyRelations:

```
- edm4eic::TrackerHit trackerHits // Hits that were used for this track
- edm4eic::Track   tracks         // Tracks (segments) that have been combined to create this track
```

Should we add a
oneToMany: cluster?