Association of SciFiClusters to MCHits and MCParticles

version 2

24 / 04 / 2014

Development notes (1)

- Various LHCb sub-detector digitizations include an intermediate step, where **Digits** and **MCDigits** are created
- A link to the relevant MCHit is stored in a MCDigit object
- When a "Cluster –> MCHit" linker table is created, the MCHits are determined through the chain:
  Cluster → Digits → MCDigits → MCHits

- The current BGV digitization is very simple, we just do **MCHit → SciFiCluster**, no intermediate steps
- Therefore, the BGV MC Links are implemented in a different way
  - No new algorithm
  - The linker table is created directly in the Digitization algorithm
    SciFiDAQ/src/DigiAlgSimple.cpp
- Eventually, when we get the "Digits" intermediate step, we can modify this code too
- Generally, one can use a chain that is "identical, but reverse" to the digitization

Development notes (2)

- A weight is assigned to each link **Cluster → MCHit**
  - The value of the MCHit energy deposit is used (as done in the Velo algorithms)

Simplification

- In the current digitization we don’t add MCHit deposits if they are in the same cluster, but reject the second occurrence
- In the future, we should allow to have more than 1 MCHits associated to a Cluster and provide links to all

Usage

Currently it is possible to

- Get the MCHit(s) or the MCParticle(s) associated to a given SciFiCluster (“forward” links)
- Get the SciFiCluster(s) associated to a given MCHit or MCParticle (“backward” links)

Example code

- The links are created in the digitization algorithm SciFiDAQ/src/DigiAlgSimple.cpp
- The created links are stored in the produced .digi file
  - E.g. /afs/cern.ch/work/p/phopchev/public/BGVDigi/DigiSamples/raw-sim1r0-geo1r16-mclinks-1000ev.digi
- An example algorithm that uses these links is available here: SciFiDAQ/src/TestMCLinkers.cpp

In the pipeline

BGV Track → MCParticle(s)

- A “track” object is needed
- The implementation should be straightforward
    - Loop over the clusters of the track
    - Get the associated MCParticles and make links with the one(s) with largest total weight