Subject: Monte Carlo Access
Posted by Sebastian Neubert on Tue, 15 May 2007 12:16:56 GMT

Hi!

I was thinking about the way to handle monte carlo information. Especially the question: Why could it be good to not put MonteCarlo info into the reco objects themselves but have associators which create maps between reco- and mctruth-objects.

The reason is a simple one: We want to use the same software for MCstudies as well as later for real data. real data has no MCtruth. so the corresponding members would be unused and initialized with some dummy value. In that case I would have to check always if there really is valid mctruth data available. A nuissance.

An associator is a good solution because it keeps reco objects free of unneeded references.

By the way: real data is not far away since we are already preparing to put TPC data from the test chamber into the framework.

Cheers! Sebastian.

Subject: Re: Monte Carlo Access
Posted by StefanoSpataro on Tue, 15 May 2007 12:28:33 GMT

Hello,

I think MC stuff is not so easy as it could appear.
As example, let's consider that your track is made by 20 points, and of course each point will have a MC id (so pid, p theta, and so on). And you wantt to store this info inside your track.

The id could be the same, but some points could come from different tracks (fake? wrong pattern recognition? noise?).

So what now, you give one MC id to your track (which is not correct, because your tracks is made by several particles, so you will have problems in computing purity efficiency and so on), or you store the MC id for all the points that contribute to the track (so 20 MC id).

You can see that both the options are not so good, with the first you loose informations, with the second you duplicate your MC id.

I think with the index to MCtrack you do not lose information, you store only one integer per point, and after you can start to elaborate algorythm to identify the real id of the particle without modifying what is streamed inside your track object.

This is only my personal point of view, I could be wrong

Hi Stefano!

I get your point. In case of the tpc it is even worse since I have to store even the event id from which a hit came. I use the classes McId and McIdCollection for this. McIdCollection has already some methods that allow such purity studies.

But that was not my point. What I want to say is, that such information should not live in the reco-track object at all!

There should be no member such as CbmMcTrack* _mctruth; in any reconstructed object.

Cheers! Sebastian.