Subject: Re: Strategy discussion on Track objects
Posted by StefanoSpataro on Fri, 01 May 2009 14:36:44 GMT
View Forum Message <> Reply to Message

Few comments from my side:

Bertram Kopf wrote Hi,
you are discussing here typical software design issues. In my point of view it would be helpful to firstly work out a -more or less- proper design, followed by dummy implementations before one then starts with concrete implementations of alorithms etc.

Indeed I am asking opinion about software design. The code and the algorythm is already implemented, working and tested with my personal design, wich is similar to what is used in HADES and CBM (if I have understood well): track (from tracking detectors) -> pidtrack (the object which stores all the informations for pid, the track + emc eloss, tof, etc.) -> microcandidate (the starting point for analysis). The selectors should be set somewhere, between pidtrack and micro candidate (which at the moment are almost similar) or after microcandidate.
I am just asking if there are ideas on how to have something better.

Bertram Kopf wrote
1. Tracking: Ralph is completely right that only one common track object should be used. The important thing is to think about how such an object should look like, what features shoult it have, if there is the necessity of abstraction, etc..

The object was designed some time ago, and now it exists also in the code: PndTrack (and PndTrackCand). One should just start to move the old objects to this new track class.

Bertram Kopf wrote
2. PID: PID has in priciple nothing to do with tracking of charged particles. It should also be possible to do PID for neutral particles: e.g. gammas, neutrons, merged pi0's, electromagnetic split-offs etc. I think, it is therefore better to decouple it from the tracking and to provide only PID specific software parts. Of course, the relevant objects should have an assoiciation to the corresponding tracks or EMC clusters.

I do not understand this point. Of course PID is different from tracking, but you can perform PID only if you have tracks. The PID object has momentum, and you ahve to use momentum + the other variables (such as emc eloss) to distinguish between particle. And only if you have tracks you can understand if one emc cluster comes from a photon or from a charged track. The, I would formulate this sentence as: PID depends on tracking. Or have I missed something?
And the track matching with pid detectors, which is a very important part of pid task, depends on track parameters. Even for the photon ID. The fact that the pid matching is done inside one directory called "lhetrack" is due to the fact that at the moment the algorythm is based on the LheTrack and not on the common PndTrack written one month ago.
The fact that genfit provides a track which will never be a base PndTrack (if I have understood well), complicates the history, because in this case the abstract class cannot be used. But I have an idea.
It could be possible to write a task, inside PndTools, which loops inside Track TCA, and

creates a new TCA with PndTrack.This will be the standard track for correlation to PID detectors.
Of course the algorythm could be moved in a new pid package, once it will be uncorrelated to "local" lhetracks.

Bertram Kopf wrote
3. The interface to the analysis should not be based on the tracks. The reconstruction should finally provide lists of particle candidates (neutrals, charged, etc.) which are then
the input for the analysis. For sure, these candidates should also provide references to the corresponding tracks or clusters including covariance matrices etc. so that also these informations are easily accessible in the analysis part. Therefore one should think about how such a candidate (RhoCandidate ?) should look like.

I think this is already inside the rho package. The user needs just momentum and pid. But the user should be able also to perform his pid, or better to perform the standard pid for his channel, which could be different from another channel.

Ideas on class design about this tracking+pid+analysis are welcome. My personal idea is already inside the code, but I would like to listen to other opinions.