
Subject: Re: Global Problems

Posted by [StefanoSpataro](#) on Wed, 03 Feb 2010 11:17:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Radek Karabowicz wrote on Wed, 03 February 2010 11:49Quote:here is what was discussed and decided:

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/Minutes05Aug2008>

see "detector type"

<http://panda-wiki.gsi.de/cgi-bin/view/Computing/PandaRootTrackModel>

see "Flags".

cheers, Soeren

I think that a lot of people are unaware of the two memos you quoted in your answer. Could we put both links in the PndDetectorList.h file as comments? Thus anyone opening the file would have links to what should really be here. Actually the link to the discussion on the next evo meeting should also come here.

I think those definitions are a bit obsolete and do not apply in the actual case. At least, DetectorID is implemented as TOF DRC TPC and so on, while fDetectorType is a mixture of DetectorID and kind of object. In particular, it is also important that MVD strips and digis have a different number, considering that they are stored in two different TCAs.

Quote:

My biggest problem was that the STT hits used number "3" as the detector/point/hit identifier and TPC uses kTpcCluster for Cluster identifier, which happens to be also equal to "3".

Looking into PndDetectorList.h I was not sure what is the goal of the fDetectorType enum. There are two values for GEM that I did not use, 7 for MVD, 3 for STT that were not used, no values for DCH. I think we should make order and decide what should be there.

STT has three different values, which are kSttPoint, kSttHit, kSttHelixHit, corresponding to 13, 14, 15 (just do cout << kSttHit << endl). If you see "3" as STT hits maybe there is some bug in STT code.

However the numbers are useless, considering that we are using an enum -> the code should contain only kXxxYyyy stuff, no integers at all. In this way one can add as many elements/detectors as he want, and we do not have to think too much when adding new stuff. And if you do not use kGemPoint and kGemHit, it does not mean that nobody is using them, considering that both the values appear in the track finding, and in kalman tasks. DCH are not present because the dch code was not merged with the actual scheme, I suppose.

Quote:

Ordering parameter

The last issue is that of the ordering parameter. I agree that the distance from point (0,0,0) makes most sense as the ordering parameter for the hits. The only problem that for some detectors (or rather for the strip detectors) this value is not easy to get. The proposition to use

radius as the ordering parameter for the STT detector may and does create problems:

TrackCand no. 1 has 13 hits.

[ihit | detid | index]

[0 | 10 | 0]

[1 | 10 | 1]

[2 | 10 | 2]

[3 | 3 | 0]

[4 | 3 | 1]

[5 | 10 | 3]

[6 | 3 | 2]

[7 | 18 | 0]

[8 | 18 | 1]

[9 | 18 | 2]

[10 | 18 | 3]

[11 | 18 | 4]

[12 | 18 | 5]

This is the second track of the first event, so it does happen rather often. Here the second number is the detectorID (10->MVD Pixel, 3->STT, 18->GEM). Naturally one would expect all the MVD hits before STT hits.

greetings,
radek

In the STT case, the problem could be solved using SttHelixHit instead of SttHit. HelixHit have x y and z, then the problem should not appear anymore. In the kalman it was done in this way.

Waiting for the next meeting for further discussions...
