Subject: Global Problems
Posted by Radoslaw Karabowicz on Tue, 02 Feb 2010 16:50:00 GMT
View Forum Message <> Reply to Message

Dear developers,

As some of you know, in the last month I focused on writing the IdealTrackMerger, that would merge ideal PndTrackCand from different detectors into global tracks, that would then be fitted. The fit would take into account all the hits belonging to one track from different detectors.

I have however got to a point from which I cannot get out without your help. Mainly due to the reason that it concerns also your code. And also because I have not taken part in the discussion about the PndDetectorList from the beginning.

To cut the long story short. We are creating PndTrackCand for different detectors, and put there hits with ptrackcand->AddHit(detectorID,hitID,orderingParameter);

So far, so good.

But what is the detectorID?

I have used the DetectorId enum from the PndDetectorList.h.
The STT is constantly using fVolumeID = 3 (set in the PndStt.cxx).
DCH also uses hardcoded info (1).
On the other hand, MVD and TPC guys are using the fDetectorType enum (9 pixelMvd, 10 stripMvd, 3 tpcluster).

I have also seen that in the meantime the fDetectorType enum grew to contain information about the detector hit type, like point, digi, cluster or hit. I do not know if that was the goal from the beginning, but will follow the idea for GEM and DCH.
Do you have any other idea? And was the plan when creating both enums.

For example, if the MC point is created and added to stack, it is given DetectorId, mostly from DetectorId enum. Why not use for example kSttPoint from fDetectorType? And if we do not use this, then why to have it at all in the fDetectorType?

I hope that this message will trigger the discussion on the subject.

Ordering parameter?

Here the situation is slightly worse. When talking about the global track merger during the last collaboration meeting in Darmstadt, we have decided to use the distance from the target as the ordering parameter. This works fine for MVD, TPC and GEM. But in case of STT and DCH is very hard to get. When updating the DCH code, I've used the mcPoint->GetLength() which is the distance the particle travelled since its creation point. It was fine, because I used ideal hit producer and ideal tracking. The STT is using radius as the ordering parameter.

Therefore I would propose that we decide now on which parameter should be used as ordering parameter in all ideal track finders of various PANDA detectors.

Does the ordering parameter have to be the same as in the realistic track finder?
I think it does not have to be the same, but should be rather similar.

The track parameters that would serve best for the purpose are the fLength or fTime members of the FairMCPoint.h

My proposition is to use the fLength as the ordering parameter in various ideal track finders, as it is the one easier accessible in the realistic track finders.

Sorry for the long message and lots of reading and greetings from snowy GSI.

Sincerely yours,
radek

Subject: Re: Global Problems
Posted by Jens Sören Lange on Tue, 02 Feb 2010 17:16:57 GMT

View Forum Message <> Reply to Message

Hi Radek,

Quote:I have also seen that in the meantime the fDetectorType enum grew to contain information about the detector hit type, like point, digi, cluster or hit. I do not know if that was the goal from the beginning, ...

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

Subject: Re: Global Problems
Posted by Tobias Stockmanns on Tue, 02 Feb 2010 18:20:57 GMT

View Forum Message <> Reply to Message

Dear Radek,

part of the confusion about the fDetectorType was created by me. I extended the enum by some additional values to uniquely identify a pnddata class by an enum in fDetectorType. This also includes for example PndTrack and PndTrackCandidate which are not realy detectors. This information is used in the MC matching to identify the type of data where the links between different data sets are pointing to. Unfortunately this is in conflict with the two flags in the TrackModel which are here combined into one flag. I am sorry for this confusion but I was not aware of the Track Model.

I think we should discuss this hole issue on the next EVO meeting. I can give there a presentation of the MC propagation classes I have written so far so it gets more clear for what this enum is used for.

Cheers,

Tobias

Subject: Re: Global Problems
Posted by Johan Messchendorp on Tue, 02 Feb 2010 22:13:56 GMT

View Forum Message <> Reply to Message

Hi,

I will put it on the agenda for next week. Can we try to define proposal(s) for the enumeration of the detector types and data classes prior to the meeting? As far as I understood Tobias, the fDetectorType has grown out of his original purpose, which would either mean we go back to the original thoughts and provide the necessary lists to accommodate the MC matching or re-consider its application completely, hence renaming it. Or are there other proposals, ideas, ... ?

Greetings,

Johan.

Subject: Re: Global Problems
Posted by StefanoSpataro on Tue, 02 Feb 2010 22:35:41 GMT

View Forum Message <> Reply to Message

Radek Karabowicz wrote on Tue, 02 February 2010 17:50Does the ordering parameter have to be the same as in the realistic track finder?
I think it does not have to be the same, but should be rather similar.
The track parameters that would serve best for the purpose are the fLength or fTime members of the FairMCPoint.h

My proposition is to use the fLength as the ordering parameter in various ideal track finders, as it is the one easier accessible in the realistic track finders.

I think ideal and realistic tracking should use exactly the same criteria, if not it become difficult to disentangle different effects due to secondaries and so on. And the length is not something that in real life can be measured. For these reasons I think the distance from 0,0,0 should be fine.

About the fDetectorType, the current structure is mandatory for lhe tracking (I think it was originally taken from there) and for the reco hit factories of the kalman task. It is not clear to me which are the problems appearing now with the current code. That enum had to say from which TClonesArray one has to take the detector hits, this is the original meaning of the enum. I think the Tobias' addenda are not harming at all, and they are still following the original idea of TCA/data object.

Therefore, I have completely not understood what should we discuss about...

Subject: Re: Global Problems
Posted by Radoslaw Karabowicz on Wed, 03 Feb 2010 10:49:51 GMT

View Forum Message <> Reply to Message

Thank you all for the quick comments. It will be best to discuss the whole PndDetectorList issue on the next evo meeting.

Quote: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.

To Stefano:
Quote:About the fDetectorType, the current structure is mandatory for lhe tracking (I think it was originally taken from there) and for the reco hit factories of the kalman task. It is not clear to me which are the problems appearing now with the current code. That enum had to say from which TClonesArray one has to take the detector hits, this is the original meaning of the enum. I think the Tobias' addenda are not harming at all, and they are still following the original idea of TCA/data object.
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.

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

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 lhe 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 lhe kalman it was done in this way.

Waiting for the next meeting for further discussions...

Subject: Re: Global Problems
Posted by Lia Lavezzi on Wed, 03 Feb 2010 11:27:27 GMT
View Forum Message <> Reply to Message

Hi,
concerning the STT detID it is still set by hand to 3 in the code: I have to change it, sorry!

For the ordering parameter I am not sure that the substitution PndSttHit with PndSttHelixHit is so easy in the track merging. As far as I understand here the idea is to merge the trackCand coming from the various track finders and for the STT the PndSttHelixHit are available only after the local reconstruction, not at track finding stage...

Ciao,
Lia.

Update: the detID is set to kSTT in svn now!

Subject: Re: Global Problems
Posted by Anonymous Poster on Wed, 03 Feb 2010 12:39:44 GMT
View Forum Message <> Reply to Message

Hi Radek,

thanks for all the work you put into this. I think that the detector ID problem will be solved in the next Evo meeting, right?

Concerning the ordering parameter:
I do not agree with the worry that the ordering parameter should be similar in MC truth and real pattern reco. They are fundamentally different things. You proposed to use the time in event parameter. I do not know how it is defined exactly, but if it provides a measurement for when the hits were generated in MC, that is what we should use. Anyway, I think you have the best overview over this issue, and I would suggest to just follow what you think is best.

I am much looking forward to have a global tracking in PandaROOT soon including all detectors!!

Cheers, Christian

Subject: Re: Global Problems
Posted by Radoslaw Karabowicz on Wed, 03 Feb 2010 12:45:15 GMT
View Forum Message <> Reply to Message

Quote:Concerning the ordering parameter:

I do not agree with the worry that the ordering parameter should be similar in MC truth and real pattern reco. They are fundamentally different things. You proposed to use the time in event parameter. I do not know how it is defined exactly, but if it provides a measurement for when the hits were generated in MC, that is what we should use. Anyway, I think you have the best overview over this issue, and I would suggest to just follow what you think is best.

On contrary, MC truth and real track finders are very similar. I can imagine cases, when you need to mix both these methods. For example, if you want to write a track finder for detector A that would use tracks found in detector B as a seed. In that case one tests it with ideal tracking in A and real tracking in B. You have the mixture where the ordering parameter should be the same.

yours

radek

ps. I am almost ready with the ideal global merger;) should have it in SVN very soon

Subject: Re: Global Problems
Posted by Lia Lavezzi on Wed, 03 Feb 2010 12:52:49 GMT
View Forum Message <> Reply to Message

I though a little about the ordering in STT.

If I remember correctly in the real track finder the hits are ordered via the 3D distance from (0, 0, 0): Gianluigi, is this correct? In fact in the real track finder an hypothesis on the reconstructed x, y, z coordinates is available.

In the ideal track finder this is not true, but what about using here the MC true coordinates xtot, ytot, ztot from the PndSttPoint and calculate the 3D distance using them instead of the x, y, z from PndSttHit (which give only the coordinates of the center of the tube)? They can play the role of the reconstructed x, y, z.

It is more or less what Stefano was suggesting with the SttHelixHit, but instead of using SttHelixHits for both finders (since as I said in the last mail we don' t have them at this stage), we use PndSttPoint coordinates in the ideal finder and reconstructed coordinates in the real one.

What do you think?
Ciao,
Lia.

P.S.: indeed tests are ongoing (as Gianluigi said at last EVO meeting). If it turns out that the local stt helix fit does not improve the track finder hypothesis on coordinates and momentum we can also decide to get rid of the local track fitting and produce the SttHelixHit directly via the track finder (or just after it)... but this has still to be decided!

Subject: Re: Global Problems
Posted by Radoslaw Karabowicz on Wed, 03 Feb 2010 13:02:35 GMT
View Forum Message <> Reply to Message

Dear Lia,

I was also thinking about using the MC point distance to (0,0,0) as the ordering parameter for STT. This is the way I did it for the DCH.

Thank you also for changing the "3" to kSTT in the PndStt.cxx. This is however not the only place when one would have to change it. The other is the creation of PndSttHits or adding hits into PndTrackCand created by your track finder. There, as far as I can see, you are using the DetectorId, which is the kSTT set in PndStt.cxx. This in fact should be changed to kSttHit. Why? Because kSTT from DetectorId is accidentally the same integer as kMVDHitsStrip from fDetectorType.

yours,
radek

Subject: Re: Global Problems
Posted by Anonymous Poster on Wed, 03 Feb 2010 13:09:12 GMT
View Forum Message <> Reply to Message

Hi,

well that is a good point. If you can get it in this consistent way, it is nice.
I was of course again thinking of looping tracks in the MVD, but maybe we push this problem back a little. We should not be held up by this now.
So, if you take the distance from 0,0,0 now, I think everybody is happy.

Christian

Subject: Re: Global Problems
Posted by Radoslaw Karabowicz on Wed, 03 Feb 2010 13:12:44 GMT
View Forum Message <> Reply to Message

I would really like to solve the issue with PndDetectorList.h before the next evo meeting in a week;)

From what I see now in the file, the fDetectorType enumerates different type of data, that are stored in different TClonesArray (like in case of MVD) to be used at a later stage (for example in GFRecoHitFactory).

Basing on this, I will probably add another values there: kGemDigi, kDchPoint, kDchDigi, kDchHit. But even without this change the name "fDetectorType" is a bit misleading. It should be rather "DataType" or something like this. The other problems I see are purely aesthetic:
a/ the enum is growing too large, too detailed and too complicated
b/ can't we really follow one naming scheme? This is the old old old story, but checking each time wether one decided to write GEM or Gem, MC or Mc, ID or Id, getX or GetX is horribly

annoying.

yours
radek

Subject: Re: Global Problems
Posted by Lia Lavezzi on Wed, 03 Feb 2010 15:09:33 GMT
View Forum Message <> Reply to Message

I' m confused

Are we saying that we forget the DetectorId list and we use only the fDetectorType list to set the detID? (... in this case I suppose I also have to change from kSTT to kSttPoint in the PndSttPoint creation...)

Cheers,
Lia.

Subject: Re: Global Problems
Posted by Radoslaw Karabowicz on Wed, 03 Feb 2010 15:31:23 GMT
View Forum Message <> Reply to Message

I guess you feel exactly how I felt at the beginning. The problem is that there are two enumarators: DetectorId and fDetectorType, which are confusing all the picture. For some things, like creation of the MCPoints, I used the DetectorId enum, MVD is using the fDetectorType and TPC volume MCid of the FairVolume where interaction took place. You implemented now what I did, but maybe we ALL should agree on one scheme beforehand, for example fDetectorType...
Well, I guess we need to wait for the meeting to have it setup.
Anyways, in my local STT, I am using kSttHit when adding hits into PndTrackCand. Works for now;)

yours,
radek

Subject: Re: Global Problems
Posted by Ralf Kliemt on Wed, 03 Feb 2010 15:46:33 GMT
View Forum Message <> Reply to Message

Hi,

The original idea was to have two enums: Detectors and data type. Since you mostly need both information the growing made the bigger enum. Keeping it to the one list we save one integer less...

Ralf.

Subject: fDetectorType
Posted by StefanoSpataro on Wed, 03 Feb 2010 17:41:52 GMT
View Forum Message <> Reply to Message

I try to explain the original meaning of fDetectorType, regardless of its name.

In the genfit reco hit factory you associates each kind of reco hits poing to a TCA with a number, and that number is that fDetectorType. The reason of many mvd types is that you have different TCA for strips and pixels, and if you do not define them properly you have no idea on where to find the corresponding hit.

In PndTrackCand fDetectorType points to the corresponding data type, this means that for stt hit it must be fSttHit, for stt helix hit it mujt be fSttHelixHit, and so on.

If you use two integers, i.e. detector + type, it does not mean that you add only one integer... First of all, you cannot threat separately mvd strip and pixels (or you ahve to define two kind of detectors). Second, if you have a track made of 50 hits, you are adding 50 integers. Moreover, you have to do a mapping from "detector"+"type" into -> data TClonesArray. This works at the moment properly with only one integer.

This is the original working definition of fDetectorType.

Subject: Re: Global Problems
Posted by Gianluigi Boca on Wed, 03 Feb 2010 21:10:09 GMT
View Forum Message <> Reply to Message

Lia Lavezzi wrote on Wed, 03 February 2010 13:52I though a little about the ordering in STT.

If I remember correctly in the real track finder the hits are ordered via the 3D distance from (0, 0, 0): Gianluigi, is this correct? In fact in the real track finder an hypothesis on the reconstructed x, y, z coordinates is available.

Yes this is correct. In the STT Real Track Finder the PARALLEL and SKEW Straw hits are ordered by increasing distance from 0,0,0.

It should be noted that in for the STT this ordering can be done
only AFTER the pattern recognition successfully found a candidate
Helix, NOT before (the STT don't have x,y,z, position for the hits).

Also in the present code ACTUALLY the ordering parameter is not the distance from the origin but instead an equivalent one (which
I believe is measured better) : the angle FI of the hit in

CYLINDRICAL coordinates, where the cylinder has the same axis of
the helix trajectory, radius equal to the radius of the helix,
and FI is increasing for negative particles going along the
helix counterclockwise when looking at the beam, and decreasing for positive particles.

Gianluigi