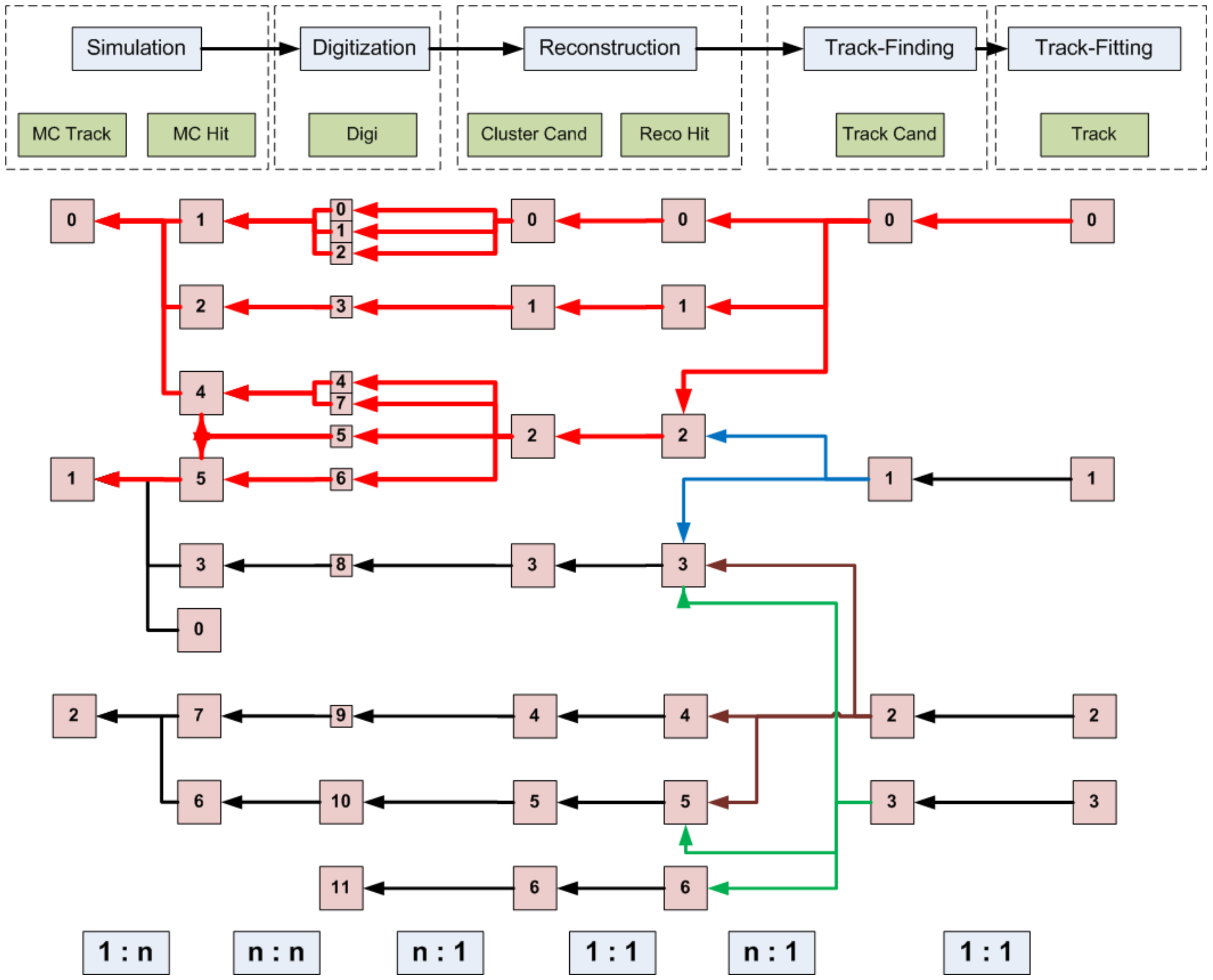
Mitglied der Helmholtz-Gemeinschaft

Distribution of MC Information

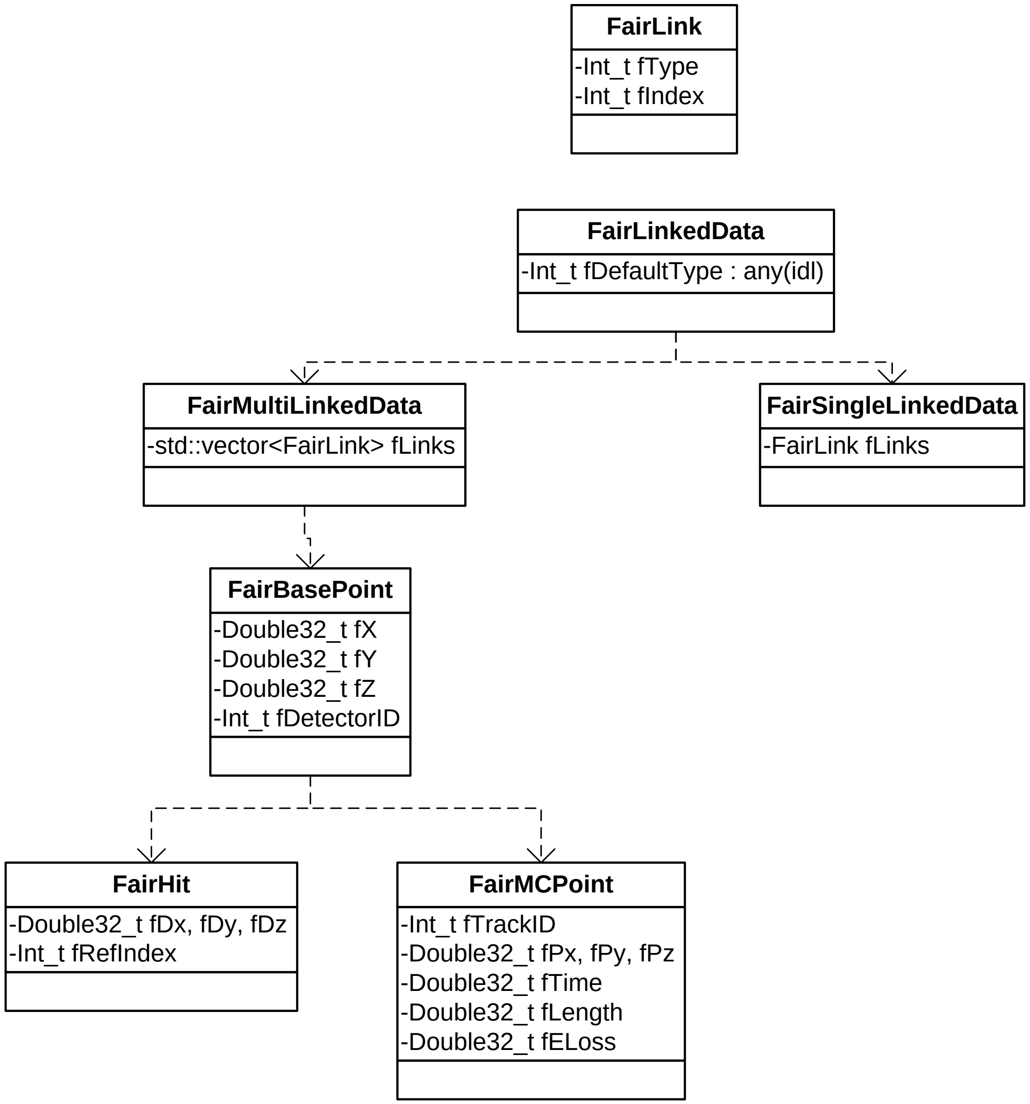
10. Februar 2010 | Tobias Stockmanns

Track match for Track 0



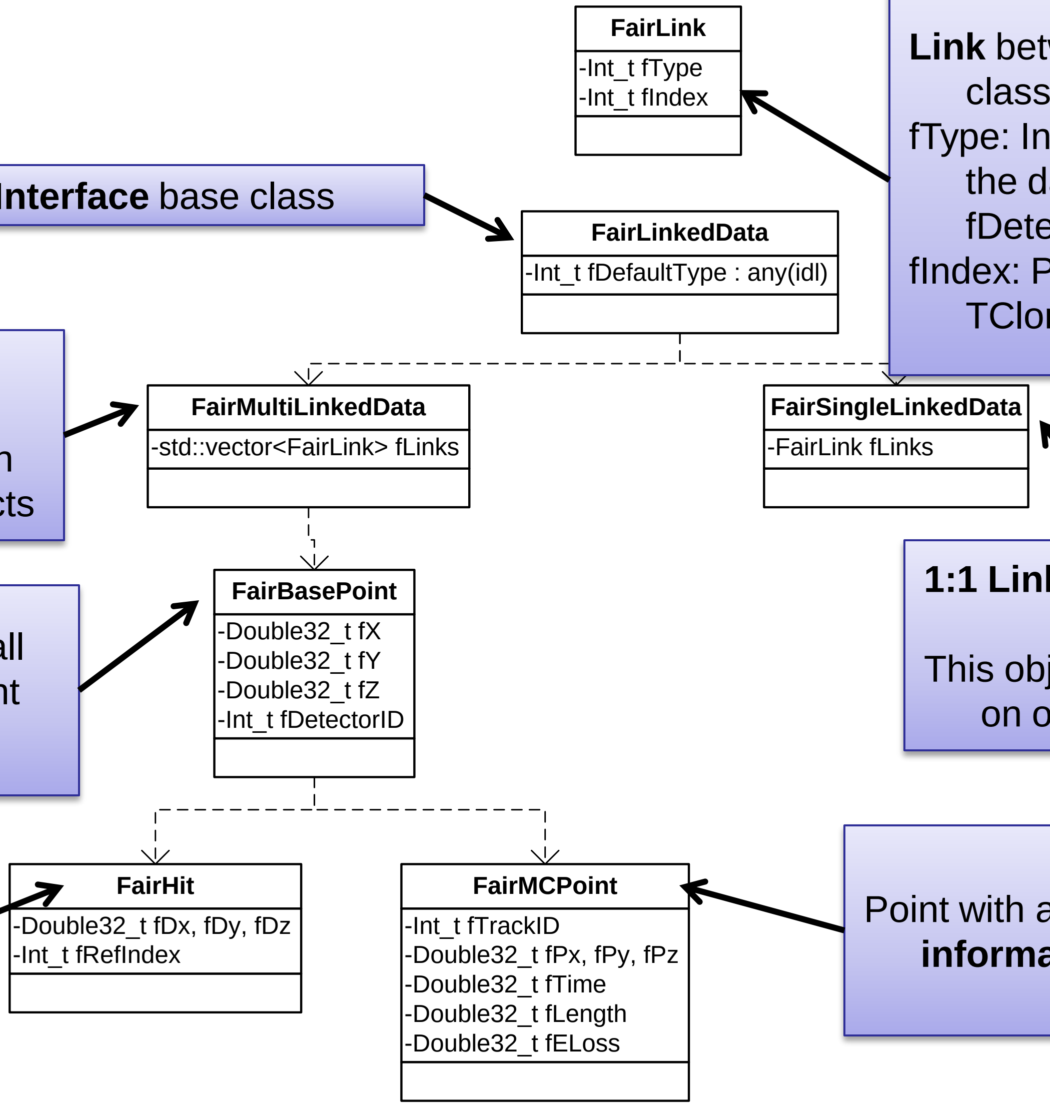
Implementation

In FairRoot:



Implementation

In FairRoot:



Link between different data classes
fType: Indicating the branch the data is stored in → fDetectorList
fIndex: Position in TClonesArray

Interface base class

1:n Link

This object is based on different data objects

1:1 Link

This object is based only on one data object

New base class for all position dependent information

Point with **errors**
fRefIndex is old link to MCPoint (should be removed)

Point with additional **MC information**

fDetectorType

```
enum fDetectorType {
        kUnknown,
        kMCTrack,
        kTpcPoint, kTpcCluster,
        kMVDPoint, kMVDDigiStrip, kMVDDigiPixel, kMVDClusterPixel, kMVDClusterStrip, kMVDHitsStrip, kMVDHitsPixel,
        kEmcCluster, kEmcBump,
        kSttPoint, kSttHit, kSttHelixHit, kSttTrackCand, kSttTrack,
        kGemPoint, kGemHit,
        kLheHit, kLheCand, kLheTrack,
        kTrackCand, kTrack
};
```

- Each entry matches uniquely to a branch in one of the root output files.
- The idea is that if you know the DetectorType (better would be DataType) you can retrieve the data from your tree via a map which links the enum with a branch name.

FairMultiLinkedData

Two important Methods:

- SetLink(FairLink link)): Clears the existing list of links and sets link as first entry
- AddLink(FairLink link, bool multi): Adds link to the vector of links. If multi is false it checks first if this link already exists and increases a weight factor for this link. In all other cases the link is added to the vector.

If you want to use MC Propagation you have to:

1. derive your classes from FairMultiLinkedData or FairHit/FairMCPoint
2. set/add the links to the data you have used to generate your actual data set

Example SttHit

```
/** Standard constructor **/
PndSttHit::PndSttHit (Int_t detID, TVector3& pos, TVector3& dpos,
                    Int_t index, Int_t flag, Double_t isochrone,
                    Double_t isochroneError, TVector3 wireDir)
  : FairHit(detID, pos, dpos, index)
{
  fIsochrone = isochrone;
  fIsochroneError = isochroneError;
  fRadial = TMath::Sqrt(pos.X() * pos.X() + pos.Y() * pos.Y());
  fWireDirection = wireDir;
  fAssigned = kFALSE;
  // stt1
  fXint = fX;
  fYint = fY;
  fZint = fZ;
  SetLink(FairLink(kSttPoint, index));
}
```

How to extract the MC Information?

Three task available:

- PndMCMatchCreatorTask:
  - runs through all available files and extracts link information from each (registered) branch
  - data is stored in a separate output file
- PndMCMatchLoaderTask:
  - reads in link file created by PndMCMatchCreatorTask
  - no other file is necessary
- PndMCMatchSelectorTask:
  - extracts information (see next slides)
  - either Creator or Loader task have to run before the Selector task

Link info for MVD

4: MVDPoint //

0: (1/3)
1: (1/3)
2: (1/3)
3: (1/3)
4: (1/2)
5: (1/2)
6: (1/2)
7: (1/1)
8: (1/1)
9: (1/1)
10: (1/1)
11: (1/0)
12: (1/0)
13: (1/0)
14: (1/0)
15: (1/0)
16: (1/22)
17: (1/22)
18: (1/22)
19: (1/22)

5: MVDStripDigis //

0: (4/2)
1: (4/2)
2: (4/2)
3: (4/3)
4: (4/3)
5: (4/3)
6: (4/6)
7: (4/6)
8: (4/6)
9: (4/6)
10: (4/9)
11: (4/9)
12: (4/10)
13: (4/10)
14: (4/10)
15: (4/10)
16: (4/14)
17: (4/14)
18: (4/14)
19: (4/14)
20: (4/15)
21: (4/15)
22: (4/15)

6: MVDPixelDigis //

0: (4/0)
1: (4/0)
2: (4/1)
3: (4/1)
4: (4/4)
5: (4/4)
6: (4/5)
7: (4/5)
8: (4/5)
9: (4/7)
10: (4/8)
11: (4/8)
12: (4/11)
13: (4/11)
14: (4/12)
15: (4/12)
16: (4/12)
17: (4/13)
18: (4/13)
19: (4/13)
20: (4/16)
21: (4/16)
22: (4/17)
23: (4/17)
24: (4/18)
25: (4/18)
26: (4/18)
27: (4/18)
28: (4/19)
29: (4/19)
30: (4/19)
31: (4/19)
32: (4/19)
33: (4/19)

7: MVDPixelClusterCand //

0: (6/0) (6/1)
1: (6/2) (6/3)
2: (6/4) (6/5)
3: (6/6) (6/7) (6/8)
4: (6/9)
5: (6/10) (6/11)
6: (6/12) (6/13)
7: (6/14) (6/15) (6/16)
8: (6/17) (6/18) (6/19) (6/20) (6/21)
9: (6/22) (6/23)
10: (6/24) (6/25) (6/26) (6/27)
11: (6/28) (6/29) (6/30) (6/31) (6/32) (6/33)

8: MVDStripClusterCand //

0: (5/16) (5/17)
1: (5/18) (5/19)
2: (5/10)
3: (5/11)
4: (5/20) (5/21)
5: (5/22)
6: (5/13) (5/12)
7: (5/14) (5/15)
8: (5/6) (5/7) (5/8)
9: (5/9)
10: (5/1) (5/0)
11: (5/2)
12: (5/4) (5/3)
13: (5/5)

9: MVDHitsStrip //

10: MVDHitsPixel //

0: (7/0)
1: (7/1)
2: (7/2)
3: (7/3)
4: (7/4)
5: (7/5)
6: (7/6)
7: (7/7)
8: (7/8)
9: (7/9)
10: (7/10)
11: (7/11)

Link info for STT

13: STTPoint //	14: STTHit //	15: SttHelixHit //
0: (1/3)	0: (13/0)	0: (14/69)
1: (1/3)	1: (13/1)	1: (14/70)
2: (1/3)	2: (13/2)	2: (14/71)
3: (1/3)	3: (13/3)	3: (14/72)
4: (1/3)	4: (13/4)	4: (14/73)
5: (1/3)	5: (13/5)	5: (14/74)
6: (1/3)	6: (13/6)	6: (14/75)
7: (1/3)	7: (13/7)	7: (14/76)
8: (1/3)	8: (13/8)	8: (14/77)
9: (1/3)	9: (13/9)	9: (14/78)
10: (1/3)	10: (13/10)	10: (14/79)
11: (1/3)	11: (13/11)	11: (14/80)
12: (1/3)	12: (13/12)	12: (14/81)
13: (1/3)	13: (13/13)	13: (14/82)
14: (1/3)	14: (13/14)	14: (14/83)
15: (1/3)	15: (13/15)	15: (14/84)
16: (1/3)	16: (13/16)	16: (14/85)
17: (1/3)	17: (13/17)	17: (14/86)
18: (1/3)	18: (13/18)	18: (14/87)
19: (1/3)	19: (13/19)	19: (14/88)
20: (1/2)	20: (13/20)	20: (14/89)
21: (1/2)	21: (13/21)	21: (14/90)
22: (1/2)	22: (13/22)	22: (14/91)
23: (1/2)	23: (13/23)	23: (14/92)
24: (1/2)	24: (13/24)	24: (14/93)
25: (1/2)	25: (13/25)	25: (14/45)
26: (1/2)	26: (13/26)	26: (14/46)
27: (1/2)	27: (13/27)	27: (14/47)
28: (1/2)	28: (13/28)	28: (14/48)
29: (1/2)	29: (13/29)	29: (14/49)
30: (1/2)	30: (13/30)	30: (14/50)
…	…	…

Link info for Lhe

21: LheCandidate //

0: (10/6) (10/7) (10/8) (15/0) (15/1) (15/2) (15/3) (15/4) (15/5) (15/6) (15/7) (15/8) (15/9) (15/10) (15/11) (15/12) (15/13) (15/14) (15/15) (15/16) (15/17) (15/18) (15/19) (15/20) (15/21) (15/22) (15/23) (15/24)

1: (15/38) (15/42) (10/4) (10/5) (15/25) (15/26) (15/27) (15/28) (15/29) (15/30) (15/31) (15/32) (15/33) (15/34) (15/35) (15/36) (15/37) (15/39) (15/40) (15/41) (15/43) (15/44) (15/45) (15/46) (15/47) (15/48)

2: (15/56) (15/57) (10/2) (10/3) (15/49) (15/50) (15/51) (15/52) (15/53) (15/54) (15/55) (15/58) (15/59) (15/60) (15/61) (15/62) (15/63) (15/64) (15/65) (15/66) (15/67) (15/68) (15/69) (15/70) (15/71) (15/72) (15/73)

3: (15/86) (15/89) (10/0) (10/1) (15/74) (15/75) (15/76) (15/77) (15/78) (15/79) (15/80) (15/81) (15/82) (15/83) (15/84) (15/85) (15/87) (15/88) (15/90)

22: LheTrack //

0: (21/0)

1: (21/1)

2: (21/2)

3: (21/3)

24: LheGenTrack //

0: (22/0)

1: (22/1)

2: (22/2)

3: (22/3)

Selector Task

21: LheCandidate //

0: (10/6) (10/7) (10/8) (15/0) (15/1) (15/2) (15/3) (15/4) (15/5) (15/6) (15/7) (15/8) (15/9) (15/10) (15/11) (15/12) (15/13) (15/14) (15/15) (15/16) (15/17) (15/18) (15/19) (15/20) (15/21) (15/22) (15/23) (15/24)

1: (15/38) (15/42) (10/4) (10/5) (15/25) (15/26) (15/27) (15/28) (15/29) (15/30) (15/31) (15/32) (15/33) (15/34) (15/35) (15/36) (15/37) (15/39) (15/40) (15/41) (15/43) (15/44) (15/45) (15/46) (15/47) (15/48)

2: (15/56) (15/57) (10/2) (10/3) (15/49) (15/50) (15/51) (15/52) (15/53) (15/54) (15/55) (15/58) (15/59) (15/60) (15/61) (15/62) (15/63) (15/64) (15/65) (15/66) (15/67) (15/68) (15/69) (15/70) (15/71) (15/72) (15/73)

3: (15/86) (15/89) (10/0) (10/1) (15/74) (15/75) (15/76) (15/77) (15/78) (15/79) (15/80) (15/81) (15/82) (15/83) (15/84) (15/85) (15/87) (15/88) (15/90)

22: LheTrack //

0: (21/0)

1: (21/1)

2: (21/2)

3: (21/3)

24: LheGenTrack //

0: (22/0)

1: (22/1)

2: (22/2)

3: (22/3)

MC Link from: LheGenTrack to MCTrack:

0: (1/0) (1/22)

1: (1/1)

2: (1/2)

3: (1/3)

Selector Task

21: LheCandidate //

0: (10/6) (10/7) (10/8) (15/0) (15/1) (15/2) (15/3) (15/4) (15/5) (15/6) (15/7) (15/8) (15/9) (15/10) (15/11) (15/12) (15/13) (15/14) (15/15) (15/16) (15/17) (15/18) (15/19) (15/20) (15/21) (15/22) (15/23) (15/24)

1: (15/38) (15/42) (10/4) (10/5) (15/25) (15/26) (15/27) (15/28) (15/29) (15/30) (15/31) (15/32) (15/33) (15/34) (15/35) (15/36) (15/37) (15/39) (15/40) (15/41) (15/43) (15/44) (15/45) (15/46) (15/47) (15/48)

2: (15/56) (15/57) (10/2) (10/3) (15/49) (15/50) (15/51) (15/52) (15/53) (15/54) (15/55) (15/58) (15/59) (15/60) (15/61) (15/62) (15/63) (15/64) (15/65) (15/66) (15/67) (15/68) (15/69) (15/70) (15/71) (15/72) (15/73)

3: (15/86) (15/89) (10/0) (10/1) (15/74) (15/75) (15/76) (15/77) (15/78) (15/79) (15/80) (15/81) (15/82) (15/83) (15/84) (15/85) (15/87) (15/88) (15/90)

22: LheTrack //

0: (21/0)
1: (21/1)
2: (21/2)
3: (21/3)

24: LheGenTrack //

0: (22/0)
1: (22/1)
2: (22/2)
3: (22/3)

MC Link from: LheGenTrack to MCTrack:
0: (1/0) (1/22) ←
1: (1/1)
2: (1/2)
3: (1/3)

This is strange!
I used an ideal track finder, thus there should not be two MC Tracks for one PndTrack

Selector Task

MC Link from:
LheGenTrack to MCTrack:
0: (1/0) (1/22)
1: (1/1)
2: (1/2)
3: (1/3)

Solution

MC Link from:
MvdPixelClusterCand to MvdPoint:
0: (4/0)
1: (4/1)
2: (4/4)
3: (4/5)
4: (4/7)
5: (4/8)
6: (4/11)
7: (4/12)
8: (4/13) (4/16) → (1/0) (1/22)
9: (4/17)
10: (4/18)
11: (4/19)

PixelClusterTask merged two MVD MC Hits from different PndTracks into one cluster

How to use the link information

```
TrackMatch:
PndTrack 0
P: 0.997445 GeV/c

Belongs to:
MCTrack 0
P: 1 GeV/c PID: 13

MCTrack 22
P: 0.0064986  GeV/c PID: 11
-------------------------------
```

```
TrackMatch for Track 1
P: 1.0012
Belongs to:
MCTrack 1
P: 1 PID: 13
-------------------------------

TrackMatch for Track 2
P: 0.988922
Belongs to:
MCTrack 2
P: 1 PID: 13
-------------------------------

TrackMatch for Track 3
P: 0.995288
Belongs to:
MCTrack 3
P: 1 PID: 13
-------------------------------
```

How to use the link information

```
void PndMCTestMomentumCompare::Exec(Option_t* opt)
{
    PndMCResult myResult = fMCMatch->GetMCInfo(kTrack, kMCTrack);          //Get track match

    for (int i = 0; i < myResult.GetNEntries(); i++)
    {
        PndTrack* myTrack = (PndTrack*)fTrack->At(i);                      //Get PndTrack

        PndMCLink myLinks = myResult.GetMCLink(i);                         //Get links belonging to PndTrack

        for (int j = 0; j < myLinks.GetNLinks(); j++) {                    //Loop over links
            if (myLinks.GetFairLink(j).GetType() == kMCTrack) {

                PndMCTrack* myMCTrack =                                    //Get MCTrack
                    (PndMCTrack*)fMCTrack->At(myLinks.GetFairLink(j).GetIndex());

                //Compare myTrack with myMCTrack
            }
        }
    }
}
```

Selector Task the other way around

MC Link from:
LheGenTrack to MCTrack:
0: (1/0) (1/22)
1: (1/1)
2: (1/2)
3: (1/3)

MC Link from:
MCTrack to LheGenTrack:
0: (8/0) (8/1) (8/4) (8/5) (17/0) (24/0)
1: (8/2) (8/3) (8/6) (8/7) (17/1) (24/1)
2: (8/8) (8/9) (17/2) (24/2)
3: (8/10) (8/11) (8/12) (8/13) (17/3) (24/3)
22: (10/9) (10/10) (10/11) (24/0)

Ideal track finder

MC Link from: MCTrack to STTHelixHit:

0: (8/0) (8/1) (8/4) (8/5) (15/0) (15/1) (15/2) (15/3) (15/4) (15/5) (15/6) (15/7) (15/8) (15/9) (15/10) (15/11) (15/12) (15/13) (15/14) (15/15) (15/16) (15/17) (15/18) (15/19) (15/20) (15/21) (15/22) (15/23) (15/24) (10/6) (10/7) (10/8)

1: (8/2) (8/3) (8/6) (8/7) (15/25) (15/26) (15/27) (15/28) (15/29) (15/30) (15/31) (15/32) (15/33) (15/35) (15/34) (15/36) (15/37) (15/39) (15/38) (15/40) (15/41) (15/42) (15/43) (15/44) (15/45) (15/46) (15/47) (15/48) (10/4) (10/5)

2: (8/8) (8/9) (15/49) (15/50) (15/51) (15/52) (15/53) (15/54) (15/55) (15/56) (15/57) (15/58) (15/59) (15/60) (15/61) (15/62) (15/63) (15/64) (15/65) (15/66) (15/67) (15/68) (15/69) (15/70) (15/71) (15/72) (15/73) (10/2) (10/3)

3: (8/10) (8/11) (8/12) (8/13) (15/74) (15/75) (15/76) (15/77) (15/78) (15/79) (15/80) (15/81) (15/82) (15/83) (15/84) (15/85) (15/86) (15/87) (15/88) (15/89) (15/90) (15/91) (15/92) (15/93) (10/0) (10/1)

22: (10/8) (10/9) (10/10) (10/11)

165: (15/94)(15/95)(15/96)

8: MvdStripCluster
10: MvdPixelHit
15: STTHelixHit

Summary and Further Steps

- Basic functionality is there
- Links by themselves are powerful tools for analysis of data
- Allows easy access to MC data combined with a high degree of flexibility
- Code is available at SVN:
  - /pandaroot/development/MCPropagation_TS

- Next steps:
  - Implement weighting of results
  - Connection to MicroCandidates (help from Klaus needed)
  - Planned code merge into trunk after Panda Meeting in March