newgeom Branch Update

Geometry

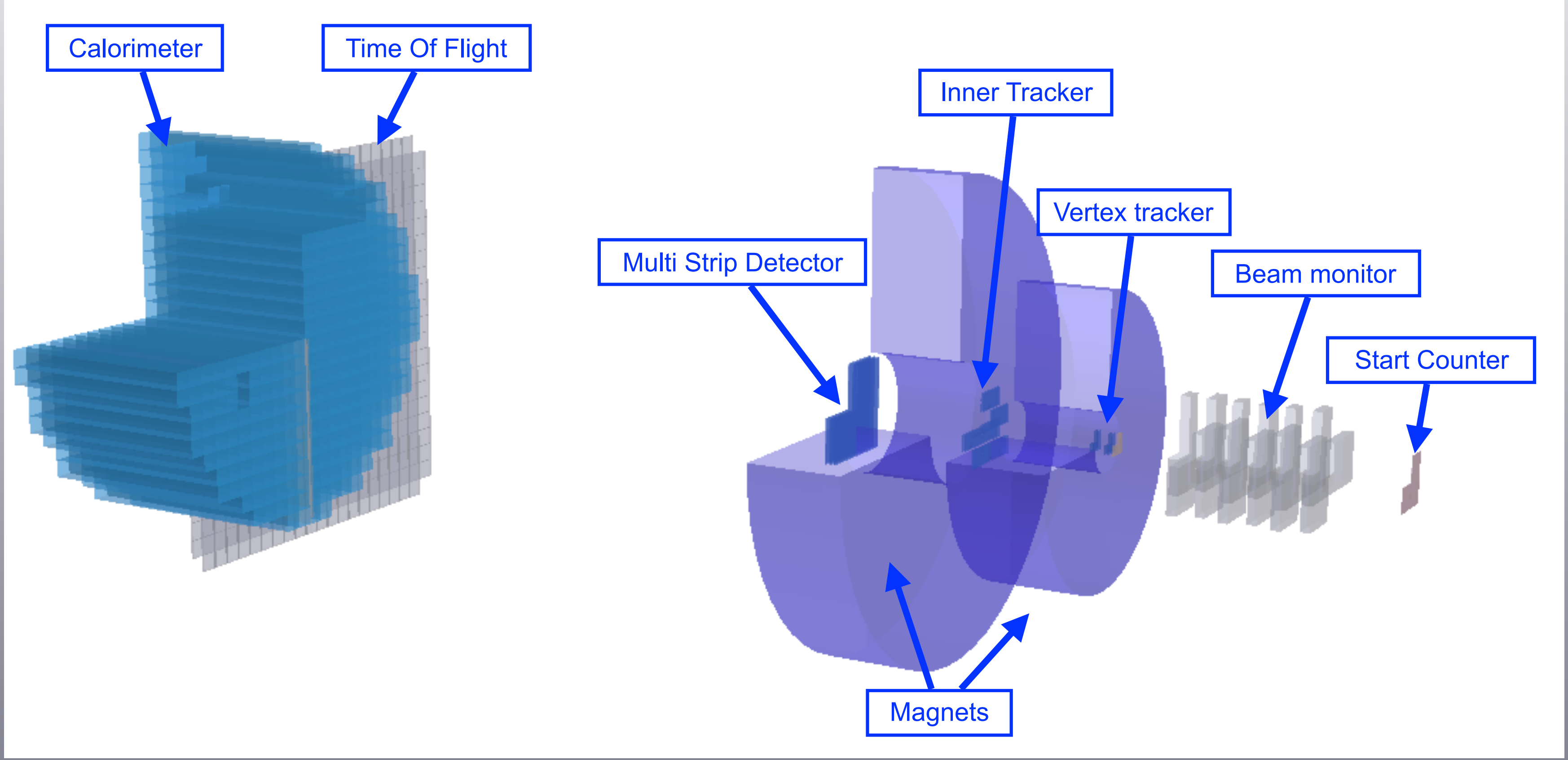
Files versionning

Debug level

Conclusions

Geometry (i)

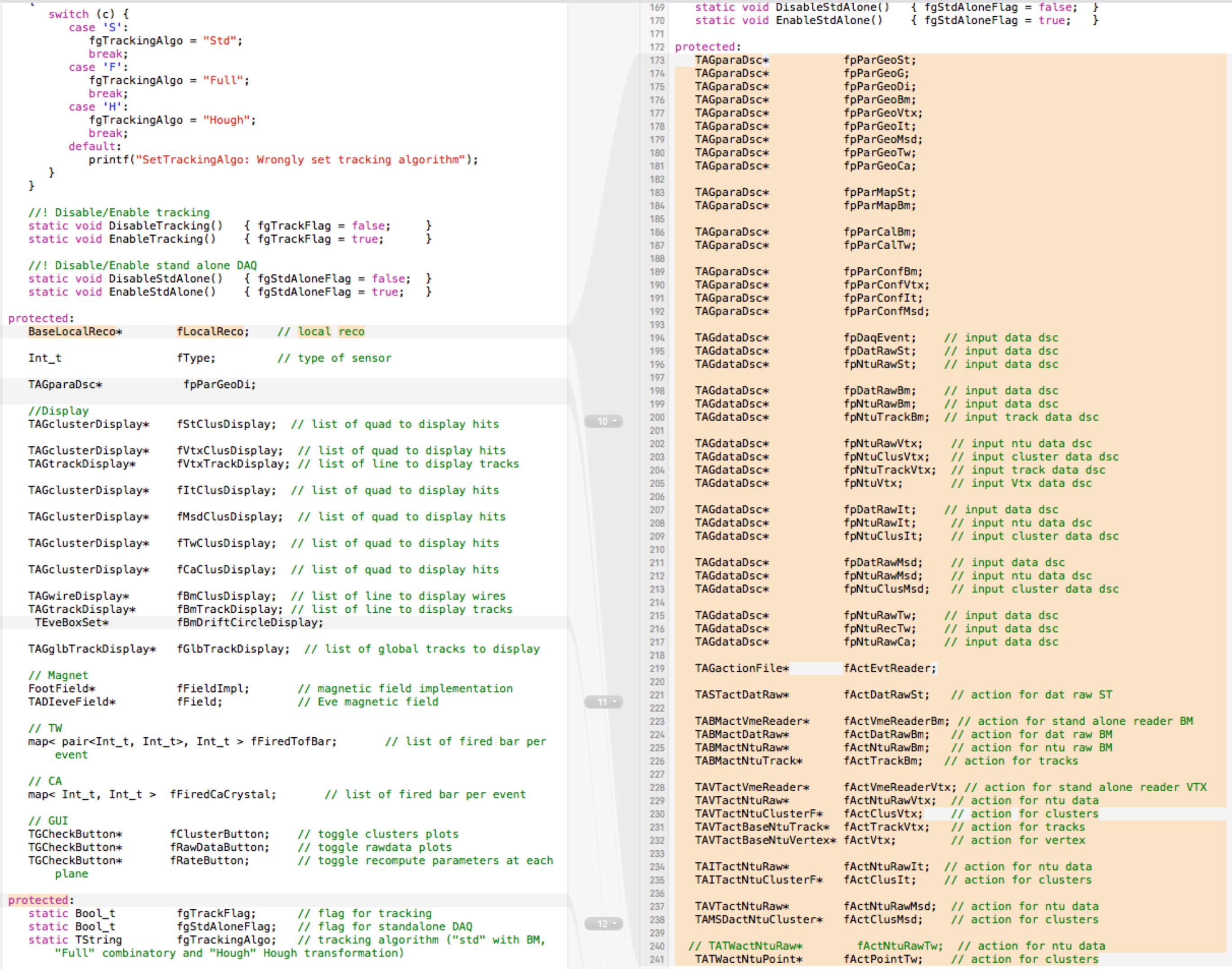
- Reconstruction:



- Full geometry of FOOT implemented

Geometry (ia)

- EventDisplay: duplicated code with LocalReco



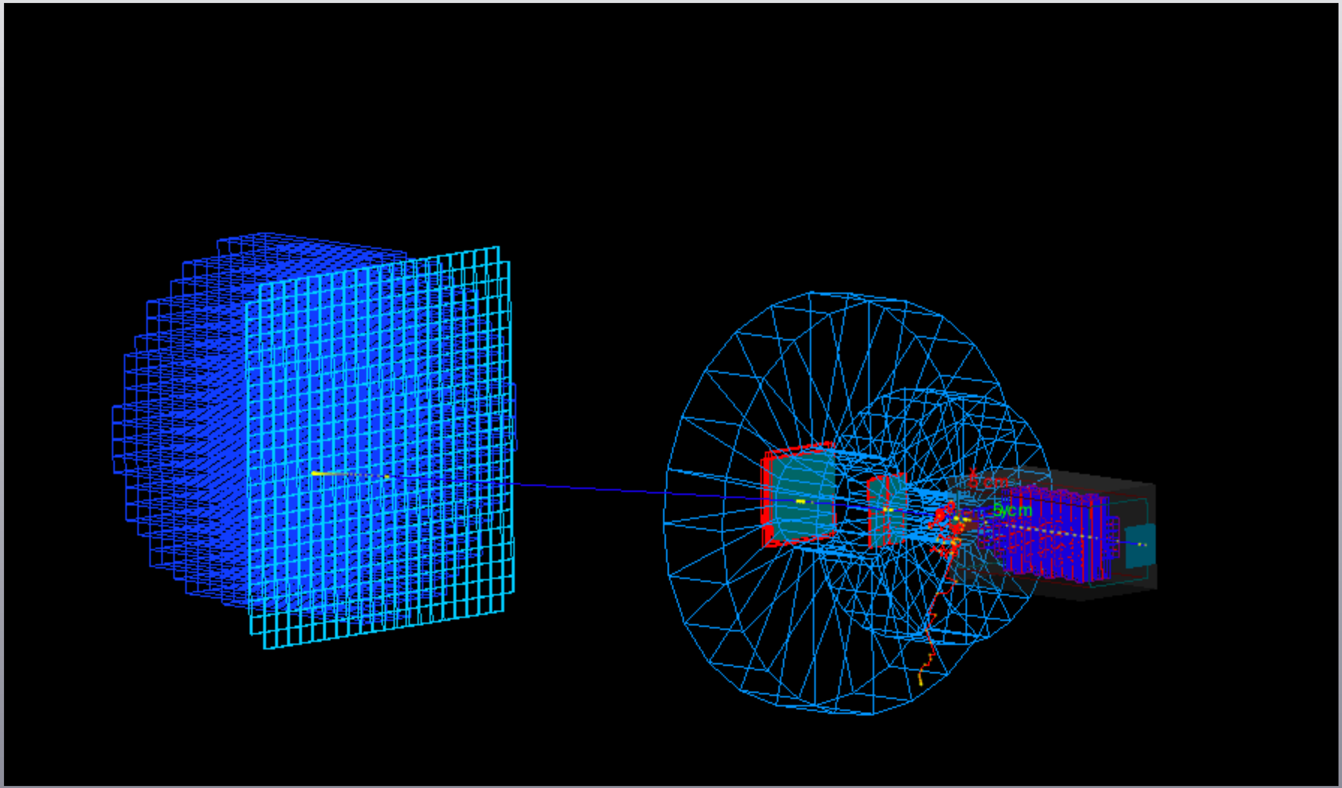
```
   switch (c) {
      case 'S':
         fgTrackingAlgo = "Std";
         break;
      case 'F':
         fgTrackingAlgo = "Full";
         break;
      case 'H':
         fgTrackingAlgo = "Hough";
         break;
      default:
         printf("SetTrackingAlgo: Wrongly set tracking algorithm");
   }
}

//! Disable/Enable tracking
static void DisableTracking()   { fgTrackFlag = false;     }
static void EnableTracking()    { fgTrackFlag = true;      }

//! Disable/Enable stand alone DAQ
static void DisableStdAlone()   { fgStdAloneFlag = false;  }
static void EnableStdAlone()    { fgStdAloneFlag = true;   }

protected:
   BaseLocalReco*        fLocalReco;    // local reco

   Int_t                 fType;         // type of sensor

   TAGparaDsc*            fpParGeoDi;

   //Display
   TAGclusterDisplay*    fStClusDisplay;  // list of quad to display hits

   TAGclusterDisplay*    fVtxClusDisplay;  // list of quad to display hits
   TAGtrackDisplay*      fVtxTrackDisplay; // list of line to display tracks

   TAGclusterDisplay*    fItClusDisplay;  // list of quad to display hits

   TAGclusterDisplay*    fMsdClusDisplay;  // list of quad to display hits

   TAGclusterDisplay*    fTwClusDisplay;  // list of quad to display hits

   TAGclusterDisplay*    fCaClusDisplay;  // list of quad to display hits

   TAGwireDisplay*       fBmClusDisplay;  // list of line to display wires
   TAGtrackDisplay*      fBmTrackDisplay; // list of line to display tracks
    TEveBoxSet*          fBmDriftCircleDisplay;

   TAGglbTrackDisplay*   fGlbTrackDisplay;  // list of global tracks to display

   // Magnet
   FootField*            fFieldImpl;       // magnetic field implementation
   TADIeveField*         fField;           // Eve magnetic field

   // TW
   map< pair<Int_t, Int_t>, Int_t > fFiredTofBar;       // list of fired bar per
       event

   // CA
   map< Int_t, Int_t >  fFiredCaCrystal;       // list of fired bar per event

   // GUI
   TGCheckButton*       fClusterButton;    // toggle clusters plots
   TGCheckButton*       fRawDataButton;    // toggle rawdata plots
   TGCheckButton*       fRateButton;       // toggle recompute parameters at each
       plane

protected:
   static Bool_t         fgTrackFlag;      // flag for tracking
   static Bool_t         fgStdAloneFlag;   // flag for standalone DAQ
   static TString        fgTrackingAlgo;   // tracking algorithm ("std" with BM,
       "Full" combinatory and "Hough" Hough transformation)
```

```
   static void DisableStdAlone()   { fgStdAloneFlag = false;  }
   static void EnableStdAlone()    { fgStdAloneFlag = true;   }

protected:
   TAGparaDsc*           fpParGeoSt;
   TAGparaDsc*           fpParGeoG;
   TAGparaDsc*           fpParGeoDi;
   TAGparaDsc*           fpParGeoBm;
   TAGparaDsc*           fpParGeoVtx;
   TAGparaDsc*           fpParGeoIt;
   TAGparaDsc*           fpParGeoMsd;
   TAGparaDsc*           fpParGeoTw;
   TAGparaDsc*           fpParGeoCa;

   TAGparaDsc*           fpParMapSt;
   TAGparaDsc*           fpParMapBm;

   TAGparaDsc*           fpParCalBm;
   TAGparaDsc*           fpParCalTw;

   TAGparaDsc*           fpParConfBm;
   TAGparaDsc*           fpParConfVtx;
   TAGparaDsc*           fpParConfIt;
   TAGparaDsc*           fpParConfMsd;

   TAGdataDsc*           fpDaqEvent;    // input data dsc
   TAGdataDsc*           fpDatRawSt;    // input data dsc
   TAGdataDsc*           fpNtuRawSt;    // input data dsc

   TAGdataDsc*           fpDatRawBm;    // input data dsc
   TAGdataDsc*           fpNtuRawBm;    // input data dsc
   TAGdataDsc*           fpNtuTrackBm;  // input track data dsc

   TAGdataDsc*           fpNtuRawVtx;    // input ntu data dsc
   TAGdataDsc*           fpNtuClusVtx;   // input cluster data dsc
   TAGdataDsc*           fpNtuTrackVtx;  // input track data dsc
   TAGdataDsc*           fpNtuVtx;       // input Vtx data dsc

   TAGdataDsc*           fpDatRawIt;    // input data dsc
   TAGdataDsc*           fpNtuRawIt;     // input ntu data dsc
   TAGdataDsc*           fpNtuClusIt;    // input cluster data dsc

   TAGdataDsc*           fpDatRawMsd;    // input data dsc
   TAGdataDsc*           fpNtuRawMsd;    // input ntu data dsc
   TAGdataDsc*           fpNtuClusMsd;   // input cluster data dsc

   TAGdataDsc*           fpNtuRawTw;    // input data dsc
   TAGdataDsc*           fpNtuRecTw;    // input data dsc
   TAGdataDsc*           fpNtuRawCa;    // input data dsc

   TAGactionFile*        fActEvtReader;

   TASTactDatRaw*        fActDatRawSt;   // action for dat raw ST

   TABMactVmeReader*     fActVmeReaderBm; // action for stand alone reader BM
   TABMactDatRaw*        fActDatRawBm;   // action for dat raw BM
   TABMactNtuRaw*        fActNtuRawBm;   // action for ntu raw BM
   TABMactNtuTrack*      fActTrackBm;   // action for tracks

   TAVTactVmeReader*     fActVmeReaderVtx; // action for stand alone reader VTX
   TAVTactNtuRaw*        fActNtuRawVtx;  // action for ntu data
   TAVTactNtuClusterF*   fActClusVtx;    // action for clusters
   TAVTactBaseNtuTrack*  fActTrackVtx;   // action for tracks
   TAVTactBaseNtuVertex* fActVtx;        // action for vertex

   TAITactNtuRaw*        fActNtuRawIt;  // action for ntu data
   TAITactNtuClusterF*   fActClusIt;    // action for clusters

   TAVTactNtuRaw*        fActNtuRawMsd;  // action for ntu data
   TAMSDactNtuCluster*   fActClusMsd;    // action for clusters

  // TATWactNtuRaw*        fActNtuRawTw;  // action for ntu data
   TATWactNtuPoint*      fActPointTw;    // action for clusters
```

- Remove local reconstruction and put a pointer to LocalReco instead

Geometry (ii)

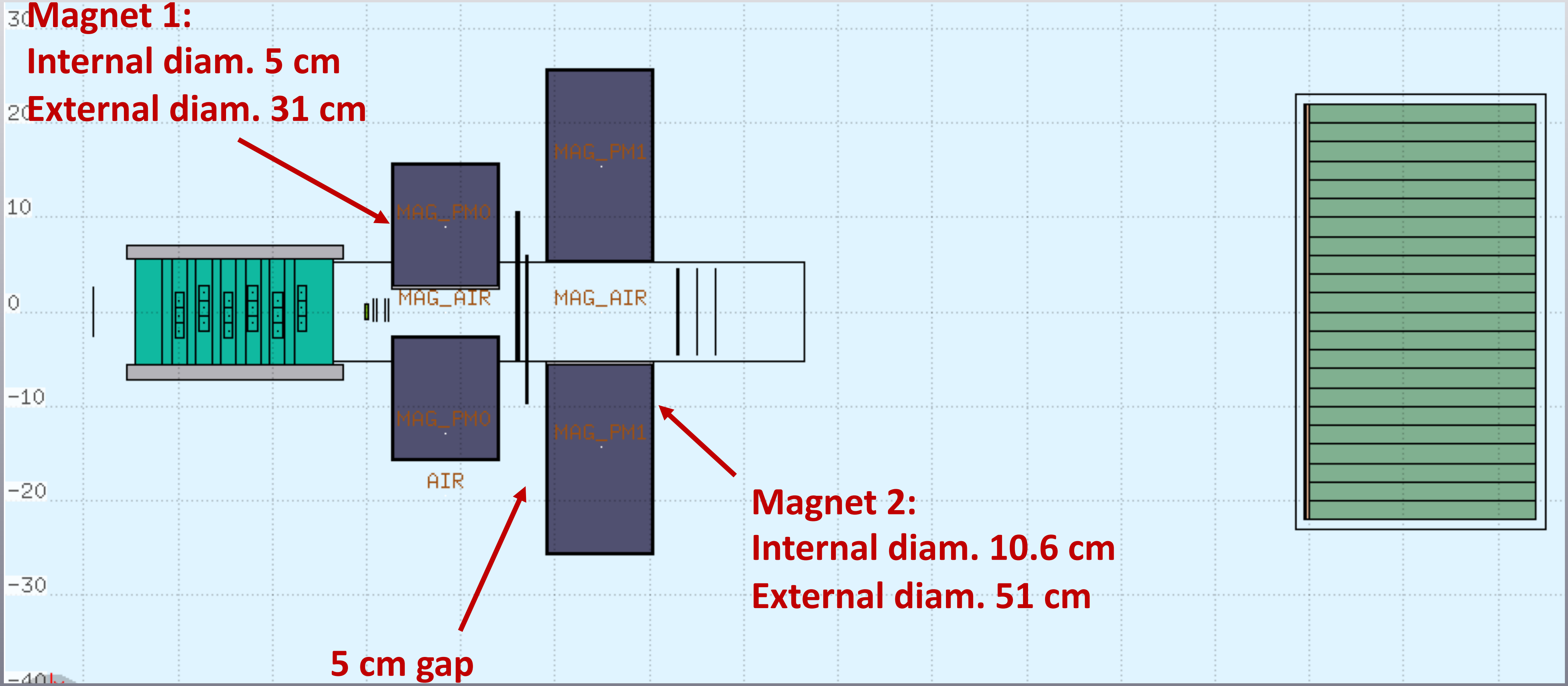
- Geant4 geometry:



- Full geometry of FOOT implemented in Geant4 (Marie)

Geometry (ii)

- Fluka geometry:



Magnet 1:
Internal diam. 5 cm
External diam. 31 cm

Magnet 2:
Internal diam. 10.6 cm
External diam. 51 cm

5 cm gap

- Works going on, not a big issue (Serena)

➡ Still missing some passive materials in MC geometry

Files versioning (i)

Campaign versioning

- We have different files for a given campaign (GSI)

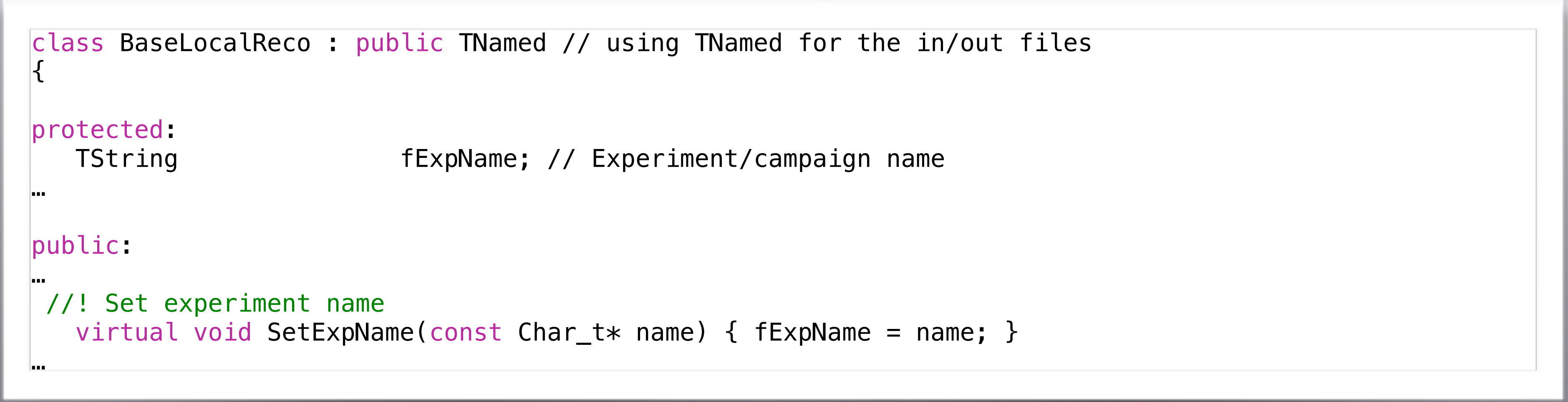
➡ We have to decide which depending or not on campaign

File	Need Versioning
TA*detector.map (geometry)	✔
TA*detector.cfg (analysis config)	✔
TA*.cal (calibration file)	✔
TA*.map (mapping file)	✗
FOOT_geo.map (global transformation)	✔

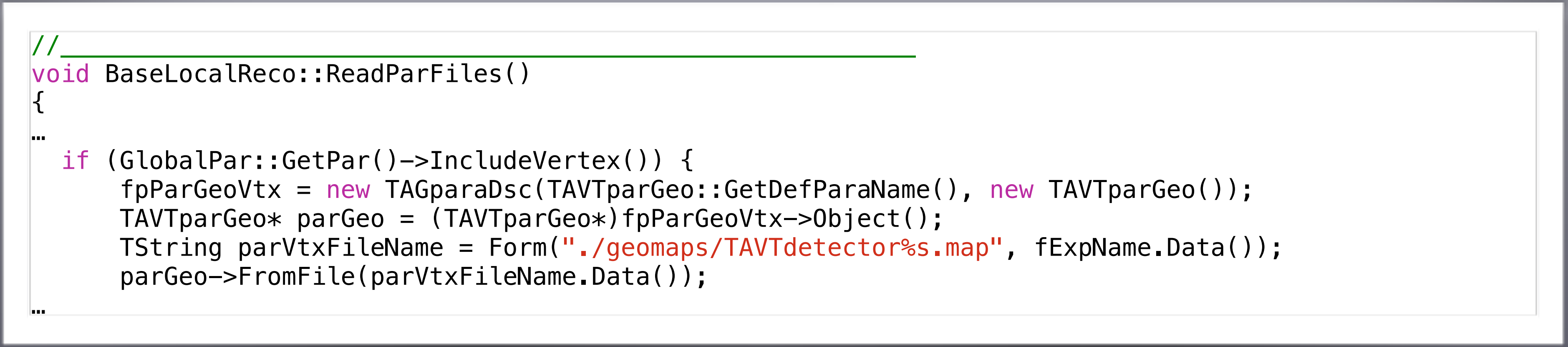
➡ Detector expert have to decide

Files versioning (ii)

- Local reconstruction
  - ➡ Add an extension to the files (e.g.: _GSI)



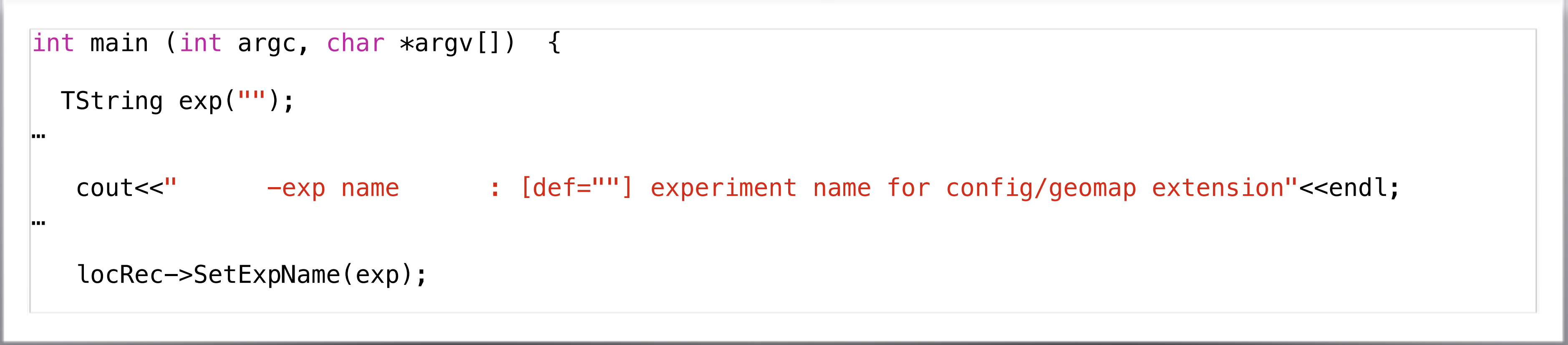
```
class BaseLocalReco : public TNamed // using TNamed for the in/out files
{

protected:
   TString               fExpName; // Experiment/campaign name
…

public:
…
 //! Set experiment name
   virtual void SetExpName(const Char_t* name) { fExpName = name; }
…
```



```
//__________________________________________________________
void BaseLocalReco::ReadParFiles()
{
…
  if (GlobalPar::GetPar()->IncludeVertex()) {
      fpParGeoVtx = new TAGparaDsc(TAVTparGeo::GetDefParaName(), new TAVTparGeo());
      TAVTparGeo* parGeo = (TAVTparGeo*)fpParGeoVtx->Object();
      TString parVtxFileName = Form("./geomaps/TAVTdetector%s.map", fExpName.Data());
      parGeo->FromFile(parVtxFileName.Data());
…
```

Files versioning (iii)

- DecodeRaw:
  - ➡ Add an extension to the files (e.g.: _GSI)



```
int main (int argc, char *argv[])  {

  TString exp("");
…
   cout<<"      -exp name      : [def=""] experiment name for config/geomap extension"<<endl;
…
   locRec->SetExpName(exp);
```

➡ Can add an extension for a given campaign for ALL files

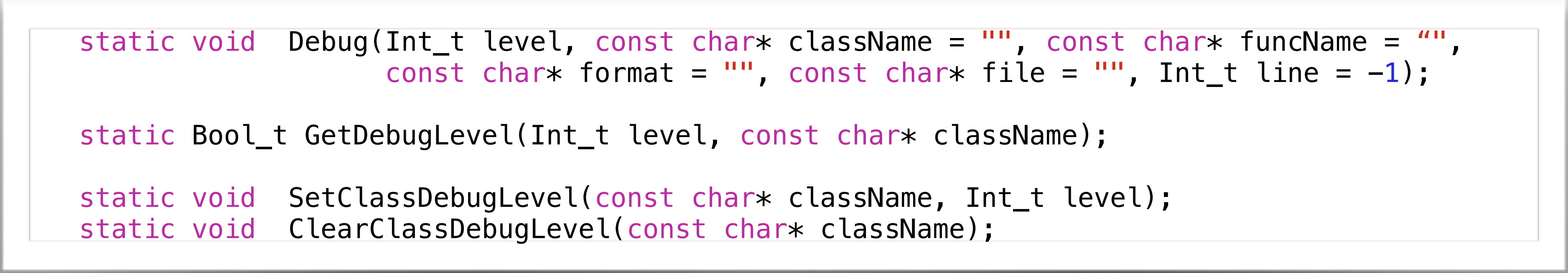
Debug Level (i)

- All action debug levels set on, hardware wise
- Lot of informations, overflowing the real errors
- Have to remove given debug level by hand and recompile

➡ Not very efficient

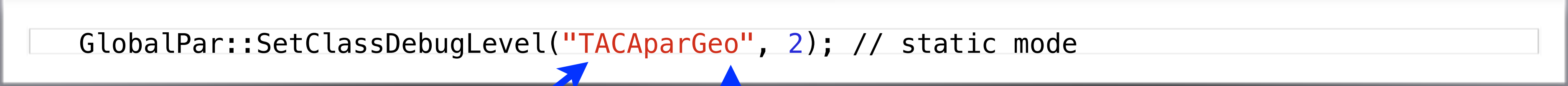
Debug Level (ii)

- Add new methods in GlobalPar class (see my previous presentation)

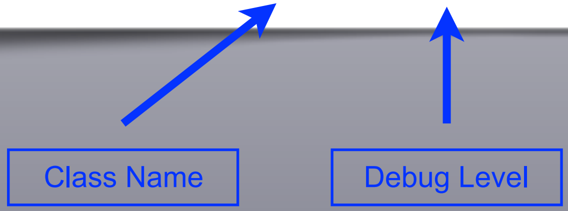


```
   static void  Debug(Int_t level, const char* className = "", const char* funcName = “",
                           const char* format = "", const char* file = "", Int_t line = -1);

   static Bool_t GetDebugLevel(Int_t level, const char* className);

   static void  SetClassDebugLevel(const char* className, Int_t level);
   static void  ClearClassDebugLevel(const char* className);
```

- Set debug level for a class



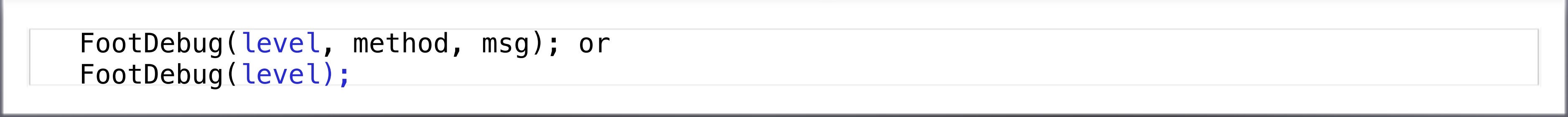
```
   GlobalPar::SetClassDebugLevel("TACAparGeo", 2); // static mode
```



Class Name

Debug Level

- Replace “fDebugLevel > level” by :



```
   FootDebug(level, method, msg); or
   FootDebug(level);
```

➡ Start with TAVT classes

Conclusions

- Full geometry with the digitizers
  - Study of the realistic performances of the setup.
  - Study with noise and intra-fragmentation in the detectors

- Versioning of file implemented

- New Debug level implemented (in progress)

➡ Need to optimize and define the setup
not only in view of resolution but also in efficiency