Updates in new_geom branch

Event display

Magnetic Field

Parameter classes…

Container classes

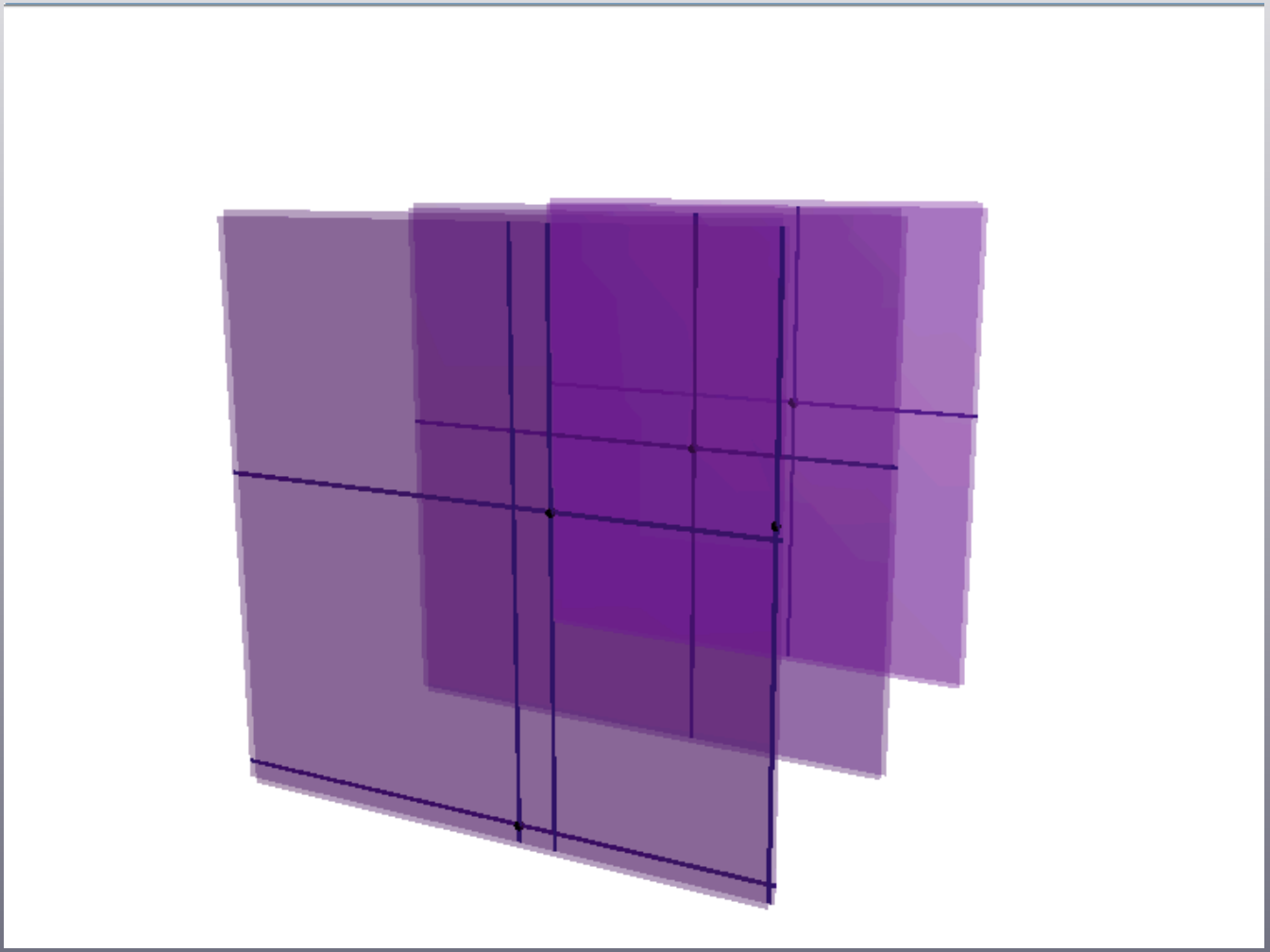
Global reconstruction

Global reconstruction Action

Conclusion

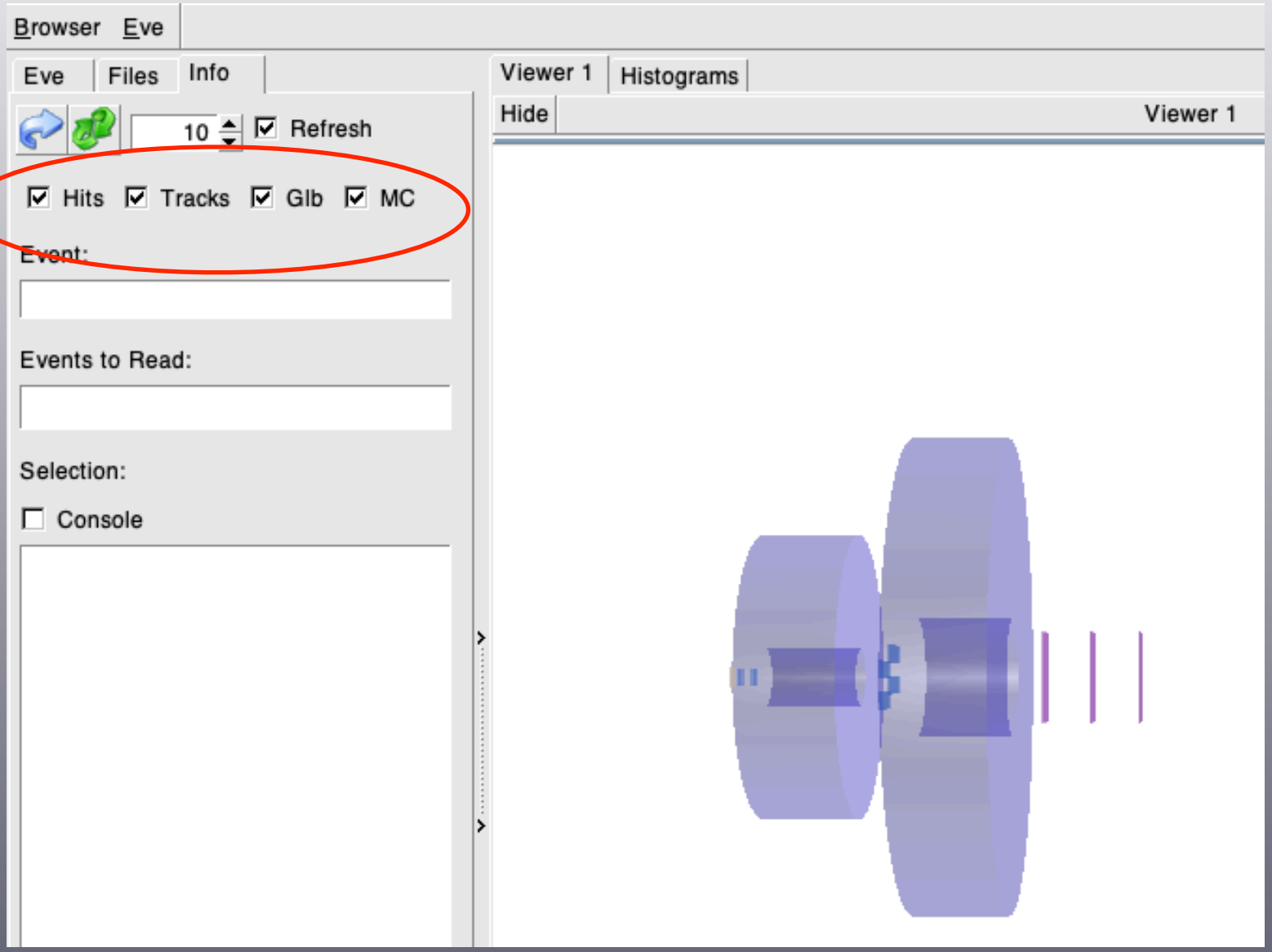
Event Display (i)

- New display for MSD (no rec point but strip)



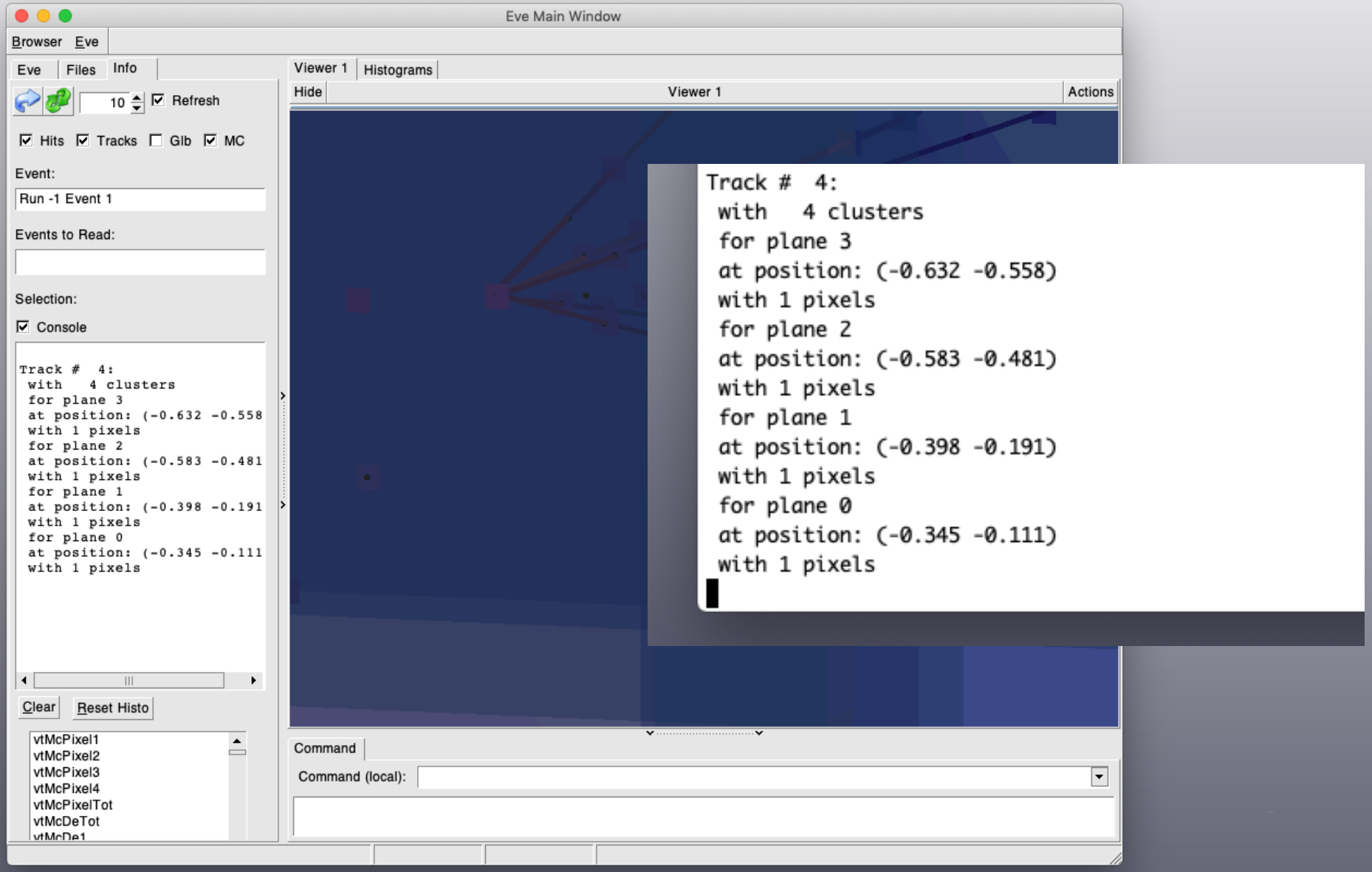
Event Display (ii)

- Add check box to toggle hits, track, global track and MC hits



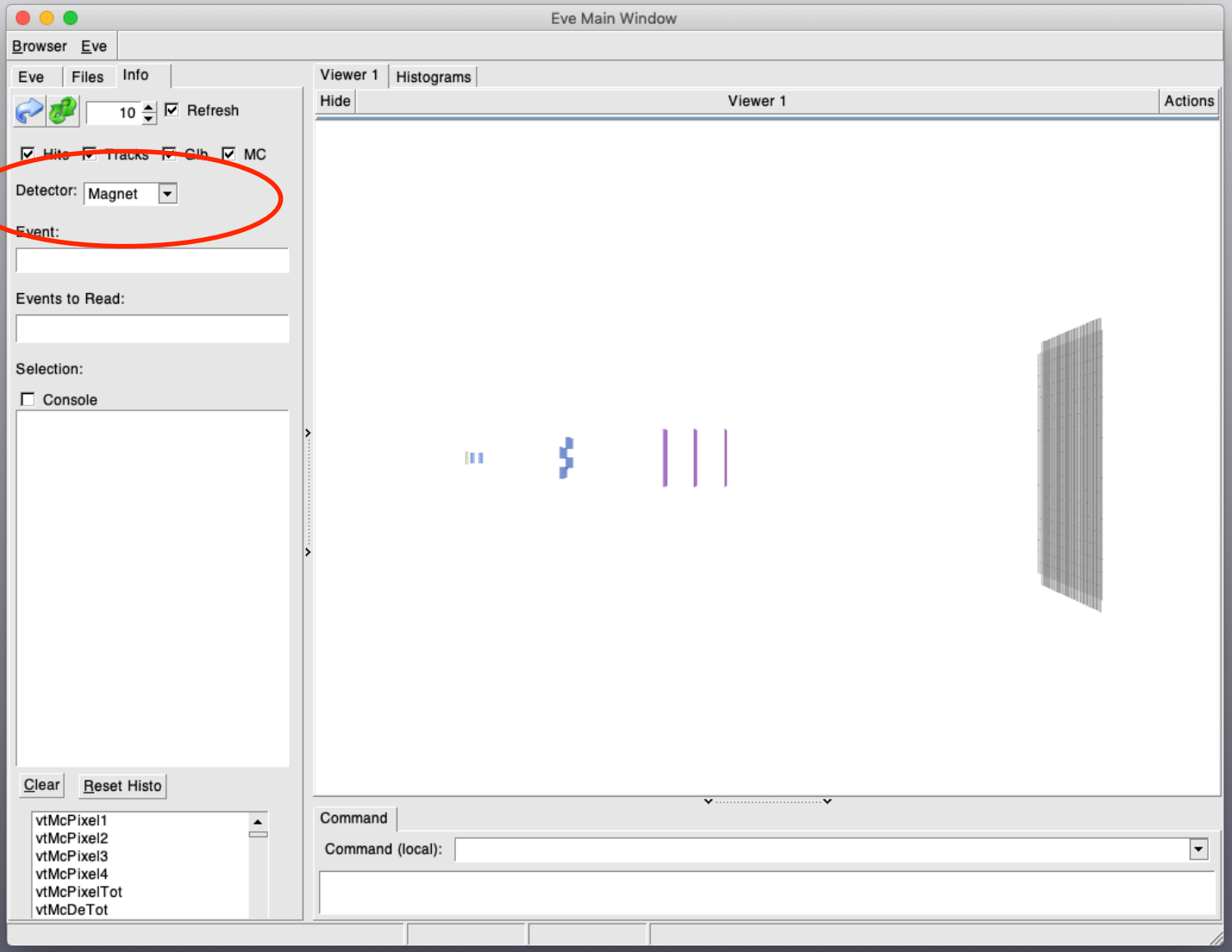
Event Display (iii)

- Add check box to printout information also on console



Event Display (iv)

- Add menu box to toggle detector drawing



Magnetic Field

- TADIgeoField: new class, based on TH3F

```
class TADIgeoField : public TVirtualMagField {

public:
    TADIgeoField(TADIparGeo* diGeo);

   ~TADIgeoField();

   TVector3 GetField(const TVector3& position)    const;
   TVector3 Interpolate(const TVector3& position) const;

   void     Field(const Double_t* pos, Double_t* fieldB);
   void     FromFile(TString& name);

private:
   TADIparGeo*  fpDiGeoMap;
   TAGgeoTrafo* fpFootGeo;

   TH3F*        fMagHistoX;
   TH3F*        fMagHistoY;
   TH3F*        fMagHistoZ;
. . .
};
```

➡ faster than FootField implementation

Parameter classes…

- Get position of a given TW layer

```
TVector3 TATWparGeo::GetLayerPosition(Int_t iLayer)
```

- Return an array of sensor Id's inside a given IT layer

```
UChar_t* TAITparGeo::GetSensorsPerLayer(Int_t iLayer)
```

- Return position in Z for a given IT layer

```
Float_t GetlayerPosZ(Int_t layer);
```

- Some bugs fixed in TAMSDntuHit, TAMSDntuRaw, TAMSDactNtuCluster, TATWactNtuMC, TATWntuHit, TATWparGeo classes

Container class

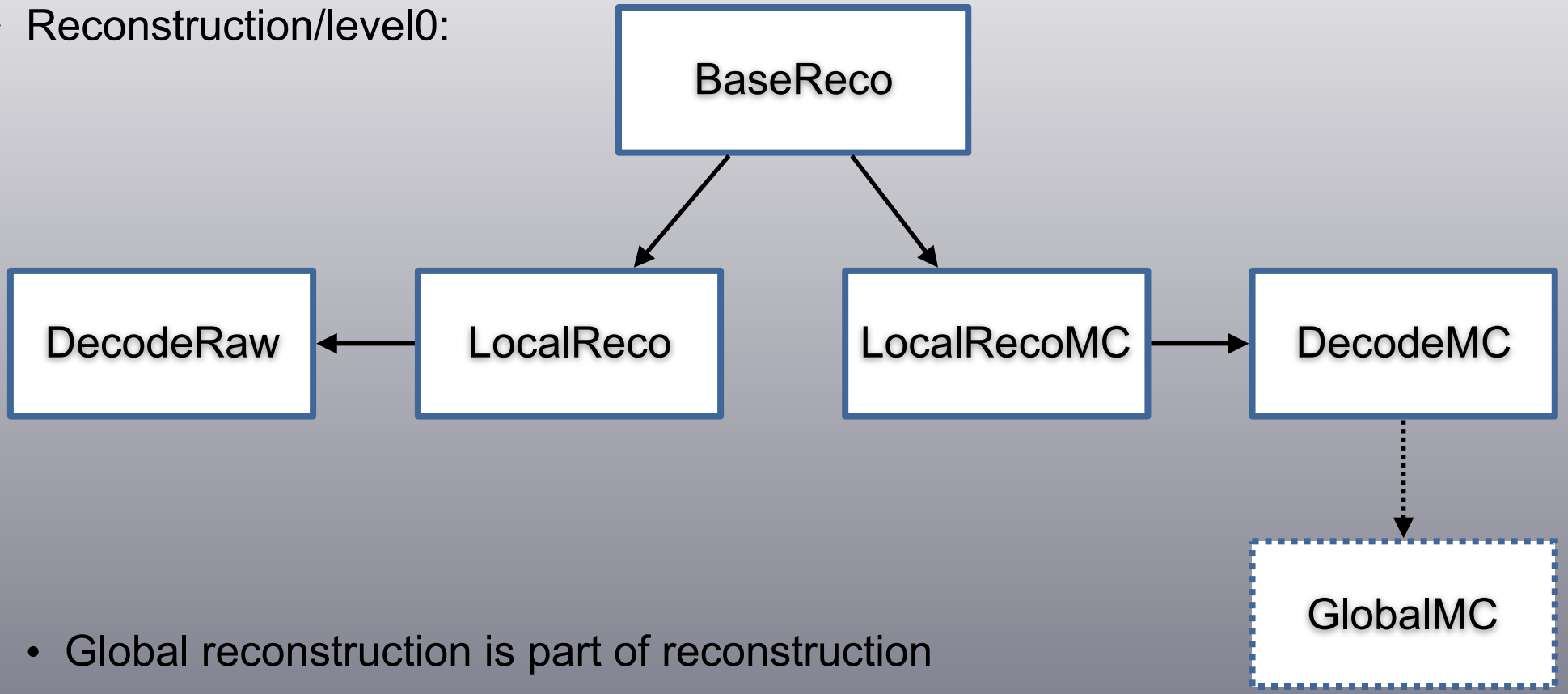
- Add position errors in TW point

```
lass TATWpoint : public TAGobject {

private:
   TVector3    m_position;      // position in detector framework
   TVector3    m_posErr;        // position error in detector framework
  . . .

public:

   TATWpoint();
   TATWpoint( double x, double dx, TATWntuHit* colHit, double y, double dy, TATWntuHit* rowHit );
   ~TATWpoint() {};

   //    All the Get methods
   TVector3  GetPosition()    const  { return m_position;           }
   TVector3  GetPosError()    const  { return m_posErr;             }

. . .
};
```

➡ Needed for covariance matrices computing (same name as for VTX/IT/MSD)

Global reconstruction (i)
(updated framework)

- Reconstruction/level0:



- Global reconstruction is part of reconstruction
- Dedicated actions for global reconstruction, activated when flag TOE on
- Possibility to perform global reco from local reco trees (GlobalToeReco class)

Global reconstruction (ii)

- GlobalToeReco: only for reconstruction from local reco **trees**

```
//_____________________________________________________________
GlobalToeReco::GlobalToeReco(TString expName, TString fileNameIn, TString fileNameout)
 : TAGobject()
{
   TAGactNtuGlbTrack::EnableStdAlone();
   fReco = new LocalRecoMC(expName, fileNameIn, fileNameout); // using MC framework to read back in Tree
}

//_____________________________________________________________
GlobalToeReco::~GlobalToeReco()
{
   delete fReco;
}
```

- Virtual interface using MC framework for reading back local reco Tree

Global reconstruction (iii)

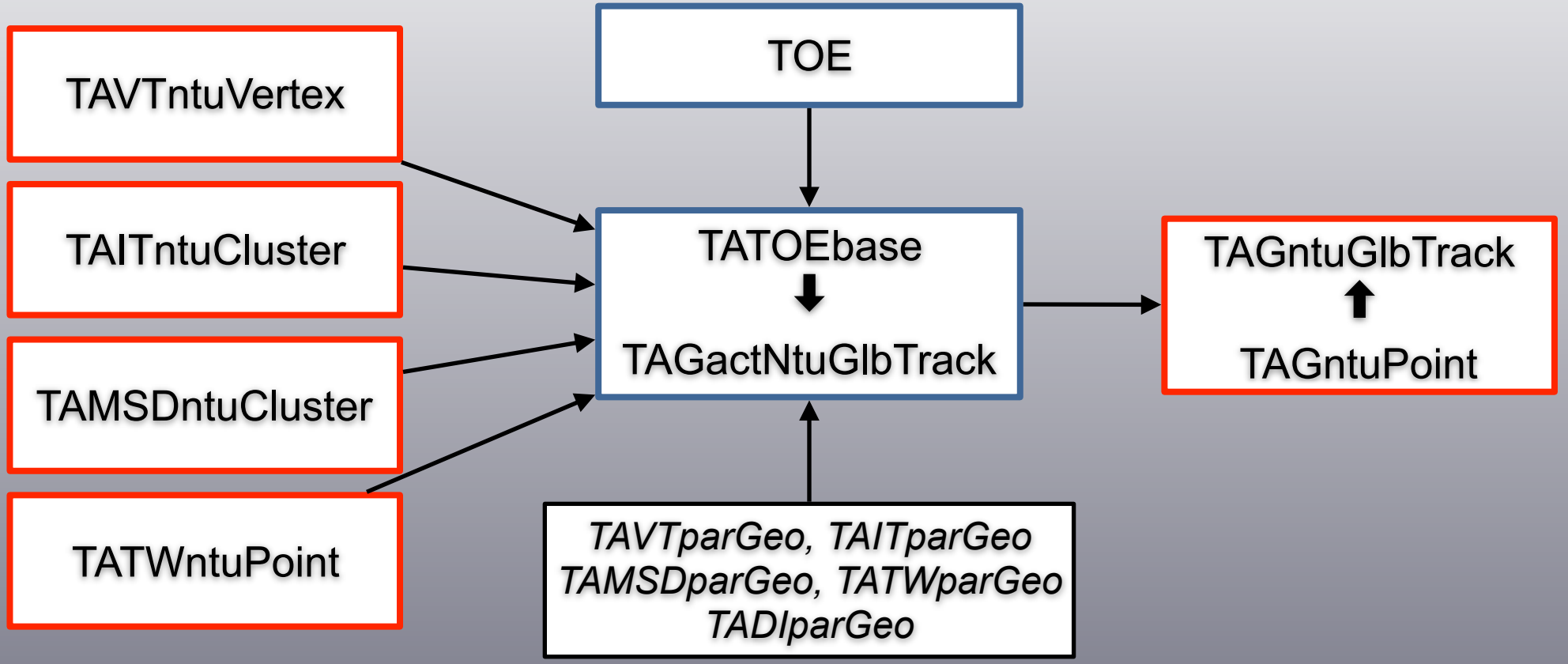
- GlobalToeReco: only for reconstruction from local reco tree

```
class GlobalToeReco : public TAGobject
{
public:
   //! default constructor
   GlobalToeReco(TString expName, TString fileNameIn = "", TString fileNameout = "");
   virtual ~GlobalToeReco();

   void EnableTree()      { fReco->EnableTree();      }
   void DisableTree()     { fReco->DisableTree();     }

   void EnableSaveHits()  { fReco->EnableSaveHits();  }
   void DisableSaveHits() { fReco->DisableSaveHits(); }

   void EnableHisto()     { fReco->EnableHisto();     }
   void DisableHisto()    { fReco->DisableHisto();    }

   void EnableTracking()  { fReco->EnableTracking();  }
   void DisableTracking() { fReco->DisableTracking(); }

   void BeforeEventLoop()       { fReco->BeforeEventLoop(); }
   void LoopEvent(Int_t nTotEv) { fReco->LoopEvent(nTotEv); }
   void AfterEventLoop()        { fReco->AfterEventLoop();  }

private:
   BaseReco*       fReco;    // local reco

   ClassDef(GlobalToeReco, 0);
};
```

- Associated DecodeGlbToe executable

Global Reconstruction Action (i)

- Scheme:



- Containers as input from VTX, IT, MSD and TW
- Container as output global tracks with global points
- TOEbase: interface between TOE and root

Global Reconstruction Action (ii)

- TAGntuGlbTrack: (unchanged)

```
class TAGntuGlbTrack : public TAGdata {

private:
   TClonesArray*    fListOfTracks;     // tracks

private:
   static TString fgkBranchName;    // Branch name in TTree

public:
   TAGntuGlbTrack();
   virtual         ~TAGntuGlbTrack();

   TAGtrack*        GetTrack(Int_t i);
   const TAGtrack*  GetTrack(Int_t i) const;
   Int_t            GetTracksN()      const;

   TClonesArray*    GetListOfTracks() { return fListOfTracks; }

   TAGtrack*        NewTrack();
   TAGtrack*        NewTrack(Double_t mass, Double_t mom, Double_t charge, Double_t tof,
                             Double_t energy, Int_t trkId);
   TAGtrack*        NewTrack(TAGtrack& track);
```

- Class adapted from FIRST

Global Reconstruction Action (iii)

- TAGtrack: add measured and corrected points in global tracks

```
class TAGtrack : public TAGobject {

public:
   TAGtrack();
   TAGtrack(Double_t mass, Double_t mom, Double_t charge, Double_t tof, Double_t energy, Int_t trkId);
   virtual         ~TAGtrack();

private:
   Double32_t       fMass;
   Double32_t       fMom;
   Double32_t       fCharge;
   Double32_t       fTof;
   Double32_t       fEnergy;
   Int_t            fTrkId;

   //Particle directions and positions computed on target middle
   TVector3         fTgtDir;
   TVector3         fTgtPos;

   //Particle directions and positions computed on ToF Wall
   TVector3         fTofPos;
   TVector3         fTofDir;

   TClonesArray*    fListOfMeasPoints;          // Attached measured points
   TClonesArray*    fListOfCorrPoints;          // Attached corrected points

   ClassDef(TAGtrack,2)
};
```

➡ To compare results for UKF

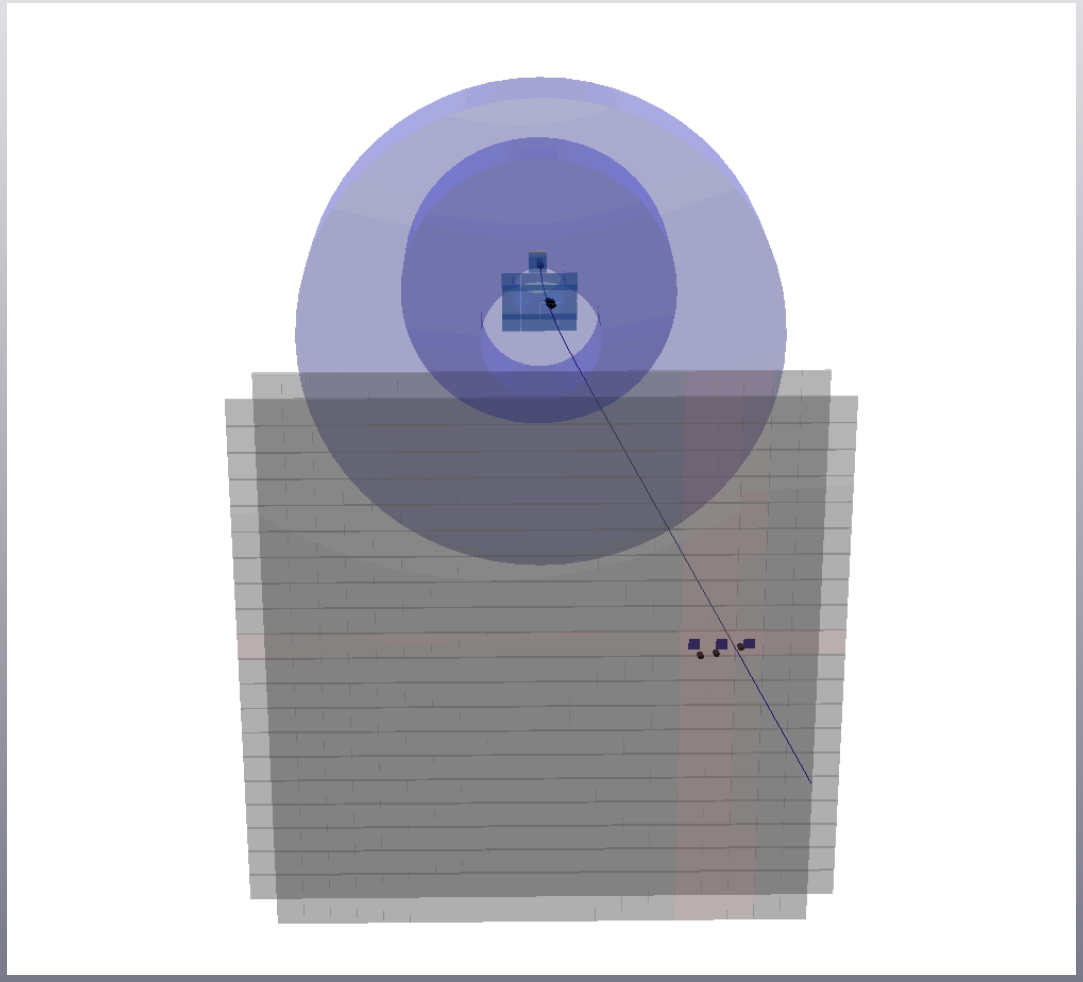
Global Reconstruction Action (iv)

- TAGpoint: remove useless members

```
class TAGpoint : public TAGobject {

private:
   TVector3    fPosition;      // position in FOOT framework
   TVector3    fPosError;      // position error in FOOT framework
   TVector3    fMomentum;      // momentum in FOOT framework
   TVector3    fMomError;      // momentum error in FOOT framework
   Int_t       fChargeZ;       // Charge Z

public:
   TAGpoint();
   TAGpoint(TVector3 pos, TVector3 posErr);
   TAGpoint(TVector3 pos, TVector3 posErr, TVector3 mom, TVector3 momErr, Int_t chargeZ = 0);
   ~TAGpoint() {};

   //    All the Get methods
   TVector3    GetPosition()         const  { return fPosition;   }
   TVector3    GetPosError()         const  { return fPosError;   }
   TVector3    GetMomentum()         const  { return fMomentum;   }
   TVector3    GetMomError()         const  { return fMomError;   }
   Int_t       GetChargeZ()          const  { return fChargeZ;    }

   void        SetPosition(TVector3 pos)    { fPosition = pos;    }
   void        SetPosError(TVector3 pos)    { fPosError = pos;    }
   void        SetMomentum(TVector3 mom)    { fMomentum = mom;    }
   void        SetMomError(TVector3 mom)    { fMomError = mom;    }
   void        SetChargeZ(Int_t z)          { fChargeZ = z;       }
   void        Clear(Option_t* opt);

   ClassDef(TAGpoint,3)
};
```

Global Reconstruction Action (v)

- TOE global track with raw data like data (except for Z charge of TW, taken from MC)



- Correct implementation of all interfaces, all algorithms in places

➡ need to tune all parameters and optimize interfaces

Conclusion

- Implemented all interfaces needed for TOE
  (needs Root compilation with -Dcxx14=ON flag)

➡ First time that we can debug the whole scheme of FOOT in new_geom branch

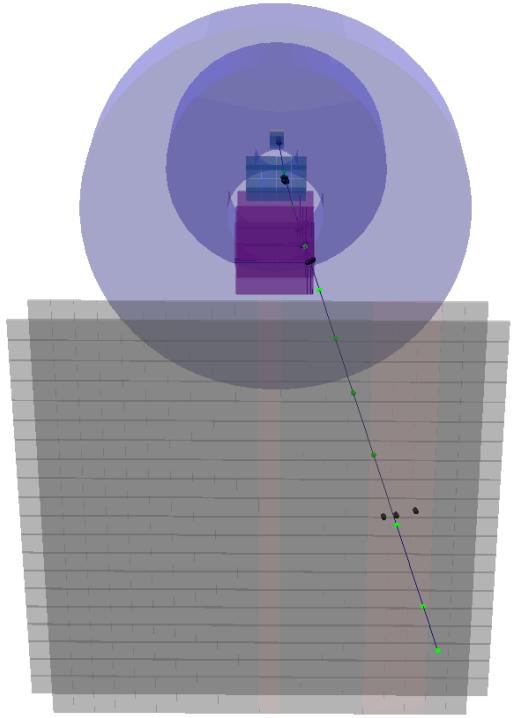
➡ First global track reconstructed with rawdata like data using TOE

Backup

Event Display (v)

- Add global track display:

Example with a MC track



➡ Correct implementation of propagation