T. Monteiro + A. Nikitenko

ECAL+HCAL tower (example code)

```
class Observing : private REO {

public:

 // the method called for each event
 void analysis(G3EventProxy * ev) {

   RecItr<CaloRU> MyRU(ev->recEvent(),"Calo_RU_Manager");
   RecItr<CaloDigi> MyDigis(ev->recEvent(),"Calo_Digi_Manager");
   RecItr<EcalPlusHcalTower> MyTower(ev->recEvent(),"Calo_Tower_Manager");

   int ntowers = 0;

   while(MyTower.next()) {

     ntowers++;

     float energy_tower = MyTower->Energy();
```

Energy from digi

```
     HepPoint3D thisPoint=MyTower->Position();
```

Position

```
     float thisTheta=thisPoint.theta();
     float thisEta=-log(tan(thisTheta*0.5));

     if(energy_tower > 1.0) {

EcalTower* ET  = MyTower->GetEcalTower();
```

*** EcalTower**

```
EcalTriggerPrimitive* ETP = ET->TriggerPrimitive();

HcalTower* HT  = MyTower->GetHcalTower();
```

*** HcalTower**

```
HcalTriggerPrimitive* HTP = HT->TriggerPrimitive();

cout << " tower= "<<MyTower->getName() ;
cout << " phi= "<< thisPoint.phi();
cout << " eta= "<< thisEta;
cout << " ETP Et= "<< ETP->Et();
cout << " HTP Et= "<< HTP->Et();
cout << " energy = "<<MyTower->Energy();
cout << " en_ecal= "<<MyTower->EnergyEcalTower();
cout << " en_hcal= "<<MyTower->EnergyHcalTower();
```

ecal tower energy
hcal tower energy

```
cout << " phi width= "<<HT->SingleOrDoublePhiTower()<<endl;
cout << " " <<endl;
```

single or double phi hcal tower

```
//test for digi energy contents

const vector<CellID> * Constituents;
```

access to constituents of ecal and hcal tower (hcal example)

```
vector<CellID>::const_iterator it;

//calculate Hcal tower energy from Digi
Constituents = HT->Constituents();
        int digi_used = 0;
double hcal_energy = 0.;
int size = Constituents->size();
for(it=Constituents->begin(); it!=Constituents->end(); it++){
  const CaloData* cd=(*it).FindCaloData("Digi");
  if(cd){
            digi_used++;
    hcal_energy += cd->Energy();
  }
}
cout <<" size= "<< size << " digi_used= " << digi_used << " hcal_energy= " << hcal_energy <<
endl;
cout << " " <<endl;


      }
    }
    cout <<" total number of towers " << ntowers << endl;
  }
private:
  void upDate(G3EventProxy * ev) { if (ev!=0) analysis(ev);}

};
```

stuff is ready be commited