```
//
// ********************************************************************
// * License and Disclaimer                                           *
// *                                                                  *
// * The  Geant4 software  is  copyright of the Copyright Holders  of *
// * the Geant4 Collaboration.  It is provided  under  the terms  and *
// * conditions of the Geant4 Software License,  included in the file *
// * LICENSE and available at  http://cern.ch/geant4/license .  These *
// * include a list of copyright holders.                             *
// *                                                                  *
// * Neither the authors of this software system, nor their employing *
// * institutes,nor the agencies providing financial support for this *
// * work  make  any representation or  warranty, express or implied, *
// * regarding  this  software system or assume any liability for its *
// * use.  Please see the license in the file  LICENSE  and URL above *
// * for the full disclaimer and the limitation of liability.         *
// *                                                                  *
// * This  code  implementation is the result of  the  scientific and *
// * technical work of the GEANT4 collaboration.                      *
// * By using,  copying,  modifying or  distributing the software (or *
// * any work based  on the software)  you  agree  to acknowledge its *
// * use  in  resulting  scientific  publications,  and indicate your *
// * acceptance of all terms of the Geant4 Software license.          *
// ********************************************************************
//
//          Filippo Ambroglini : filippo.ambroglini@pg.infn.it
//
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

#include "HEPDBoxConstructionConfig4.hh"


#include "G4NistManager.hh"
#include "HEPDSWMaterial.hh"
#include "G4Box.hh"
#include "G4SubtractionSolid.hh"
#include "G4LogicalVolume.hh"
#include "G4PVPlacement.hh"
#include "G4PVReplica.hh"

#include "G4GeometryManager.hh"
#include "G4PhysicalVolumeStore.hh"
#include "G4LogicalVolumeStore.hh"
#include "G4SolidStore.hh"
#include "G4VisAttributes.hh"
#include "G4SDManager.hh"

#include "G4UnitsTable.hh"
#include "G4PhysicalConstants.hh"
#include "G4SystemOfUnits.hh"
#include <iomanip>

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

HEPDBoxConstructionConfig4::HEPDBoxConstructionConfig4()
  :fSolidBlanket(0),fSolidWallExternal(0),fSolidWallHoneyComb(0),
   fLogicBlanket(0),fLogicWallExternal(0),fLogicWallHoneyComb(0),
   fPhysiBlanket(0),fPhysiWallExternalIn(0),fPhysiWallHoneyComb(0),fPhysiWallExt
ernalOut(0)
{
  pMaterial     = new HEPDSWMaterial();
  fBlanket_X  = 254*mm;
  fBlanket_Y  = 238*mm;
  fBlanket_Z  = 0.1*mm;

  fWallExternal_X = 345*mm;
  fWallExternal_Y = 490*mm;
  fWallExternal_Z = 0.5*mm;
```



```
  fWallHoneyComb_X = 345*mm;
  fWallHoneyComb_Y = 490*mm;
  fWallHoneyComb_Z = 5*mm;

  fWallHole_X = 237*mm;
  fWallHole_Y = 218*mm;
  fWallHole_Z = 6*mm;


  ComputeObjectsPositioning();

  // materials
  blanketMaterial        = "mylar";
  wallMaterial           = "Aluminium";
  wallHoneyCombMaterial  = "HCAluminium";
}

//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......
HEPDBoxConstructionConfig4::~HEPDBoxConstructionConfig4()
{
  if (pMaterial) delete pMaterial;
}

void HEPDBoxConstructionConfig4::ComputeObjectsPositioning(){

  transWallHole_X = fWallHoneyComb_X/2.-fWallHole_X/2.-68.5*mm;
  transWallHole_Y = fWallHoneyComb_Y/2.-fWallHole_Y/2.-44*mm;

  fPhysiBlanket_Z = 370.57*mm;

  fPhysiWallExternalIn_X = +14.5*mm;
  fPhysiWallExternalIn_Y = -92*mm;
  fPhysiWallExternalIn_Z = fPhysiBlanket_Z+fBlanket_Z/2.+fWallExternal_Z/2.;

  fPhysiWallHoneyComb_X = +14.5*mm;
  fPhysiWallHoneyComb_Y = -92*mm;
  fPhysiWallHoneyComb_Z = fPhysiWallExternalIn_Z+fWallExternal_Z/2.+fWallHoneyCo
mb_Z/2.;

  fPhysiWallExternalOut_X = +14.5*mm;
  fPhysiWallExternalOut_Y = -92*mm;
  fPhysiWallExternalOut_Z = fPhysiWallHoneyComb_Z+fWallHoneyComb_Z/2.+fWallExter
nal_Z/2.;
}


void HEPDBoxConstructionConfig4::SetBlanketMaterial(G4String aMat){
  blanketMaterial=aMat;
}
void HEPDBoxConstructionConfig4::SetWallMaterial(G4String aMat1,G4String aMat2){
  wallMaterial=aMat1;
  wallHoneyCombMaterial=aMat2;
}
//....oooOOOOOooo........oooOOOOOooo........oooOOOOOooo........oooOOOOOooo......

void HEPDBoxConstructionConfig4::Builder(G4VPhysicalVolume* motherVolume)
{

  pMaterial->DefineMaterials();
  G4Material* blanketMat = pMaterial->GetMaterial(blanketMaterial);
  G4Material* wallMat = pMaterial->GetMaterial(wallMaterial);
  G4Material* wallHCMat = pMaterial->GetMaterial(wallHoneyCombMaterial);


  G4RotationMatrix* myRot = new G4RotationMatrix;

  fSolidBlanket = new G4Box("fSolidThermalBlanket",fBlanket_X/2.,fBlanket_Y/2.,fBlank
et_Z/2);
```

```
  G4ThreeVector transWallHole(transWallHole_X,transWallHole_Y,0);
  fSolidWallExternal = new G4SubtractionSolid("fSolidWallExternal",
                                              new G4Box("WallExternal",fWallExterna
l_X/2.,fWallExternal_Y/2.,fWallExternal_Z/2.),
                                              new G4Box("WallHole",fWallHole_X/2.
,fWallHole_Y/2.,fWallHole_Z/2.),
                                              myRot,transWallHole);

  fSolidWallHoneyComb = new G4SubtractionSolid("fSolidWallHoneyComb",
                                              new G4Box("WallHoneyComb",fWallHone
yComb_X/2.,fWallHoneyComb_Y/2.,fWallHoneyComb_Z/2.),
                                              new G4Box("WallHole",fWallHole_X/2.
,fWallHole_Y/2.,fWallHole_Z/2.),
                                              myRot,transWallHole);




  fLogicBlanket = new G4LogicalVolume(fSolidBlanket,blanketMat,"fLogicThermalBlanket
");
  fLogicWallExternal = new G4LogicalVolume(fSolidWallExternal,wallMat,"fLogicWallE
xternal");
  fLogicWallHoneyComb = new G4LogicalVolume(fSolidWallHoneyComb,wallHCMat,"fLogic
WallHoneyComb");


  fPhysiBlanket = new G4PVPlacement(0,
                                    G4ThreeVector(0,0,fPhysiBlanket_Z),
                                    "HEPDBoxThermalBlanket",
                                    fLogicBlanket,
                                    motherVolume,
                                    false,0,true);

  fPhysiWallExternalIn = new G4PVPlacement(0,
                                    G4ThreeVector(fPhysiWallExternalIn_X,fPhysiW
allExternalIn_Y,fPhysiWallExternalIn_Z),
                                    "HEPDBoxWallExternalIn ",
                                    fLogicWallExternal,
                                    motherVolume,
                                    false,0,true);

  fPhysiWallHoneyComb = new G4PVPlacement(0,
                                          G4ThreeVector(fPhysiWallHoneyComb_X,fP
hysiWallHoneyComb_Y,fPhysiWallHoneyComb_Z),
                                    "HEPDBoxHoneyComb",
                                    fLogicWallHoneyComb,
                                    motherVolume,
                                    false,0,true);

  fPhysiWallExternalOut = new G4PVPlacement(0,
                                    G4ThreeVector(fPhysiWallExternalOut_X,fPhysi
WallExternalOut_Y,fPhysiWallExternalOut_Z),
                                    "HEPDBoxWallExternalOut ",
                                    fLogicWallExternal,
                                    motherVolume,
                                    false,0,true);

  //Visualization Attribute

  G4VisAttributes* attCyan = new G4VisAttributes(G4Colour::Cyan());
  attCyan->SetVisibility(true);
  attCyan->SetForceAuxEdgeVisible(true);
  fLogicWallHoneyComb->SetVisAttributes(attCyan);

  G4VisAttributes* attRed = new G4VisAttributes(G4Colour::Red());
  attRed->SetVisibility(true);
  attRed->SetForceAuxEdgeVisible(true);
  fLogicWallExternal->SetVisAttributes(attRed);
```



```
  G4VisAttributes* attGray = new G4VisAttributes(G4Colour::Gray());
  attGray->SetVisibility(true);
  attGray->SetForceAuxEdgeVisible(true);
  fLogicBlanket->SetVisAttributes(attGray);


}
```