Status of Geant4 simulation in SHOE

Marie Vanstalle, Christian Finck

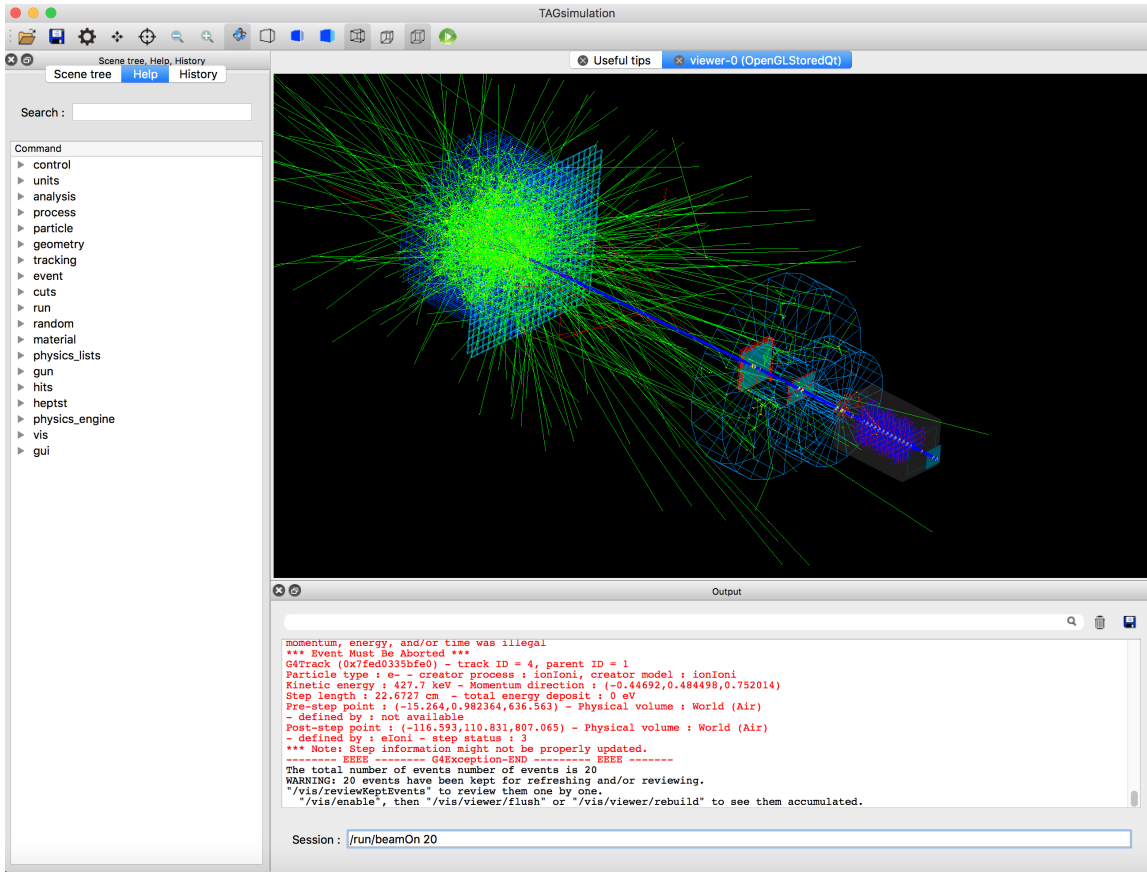
IPHC (Strasbourg)

How to use Geant4 in SHOE ?

- **After installing Geant4** (version >10), you need to re-compile SHOE adding Geant4 in the cmake : `cmake ../trunk -DGeant4_DIR=G4BUILD -DCMAKE_BUILD_TYPE=Debug` where `G4BUILD` corresponds to the path to the build directory of your Geant4 installation.
- Geant4 is located in newgeom branch, in trunk/G4simulation directory.
- After Geant4 has compiled, you can run **TAGsimulation** with different options :
  - no option : visualization mode,
  - -**out** rootFileName.root, to rename the output file,
  - -**b**, to run simulation without visualization,
  - -**seed** seedN, to change the seed number for randomization,
  - -**phys** physListName, to change the physics list.

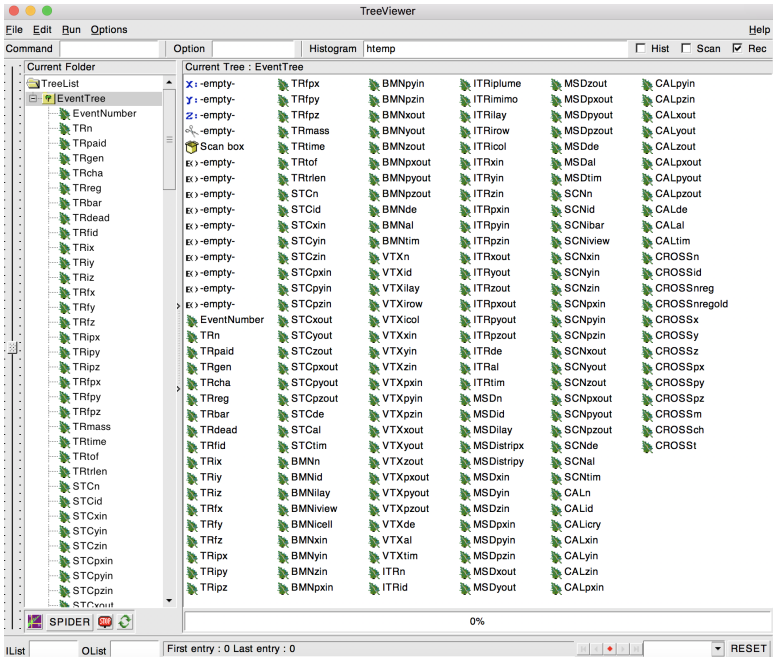
Visualization on Geant4

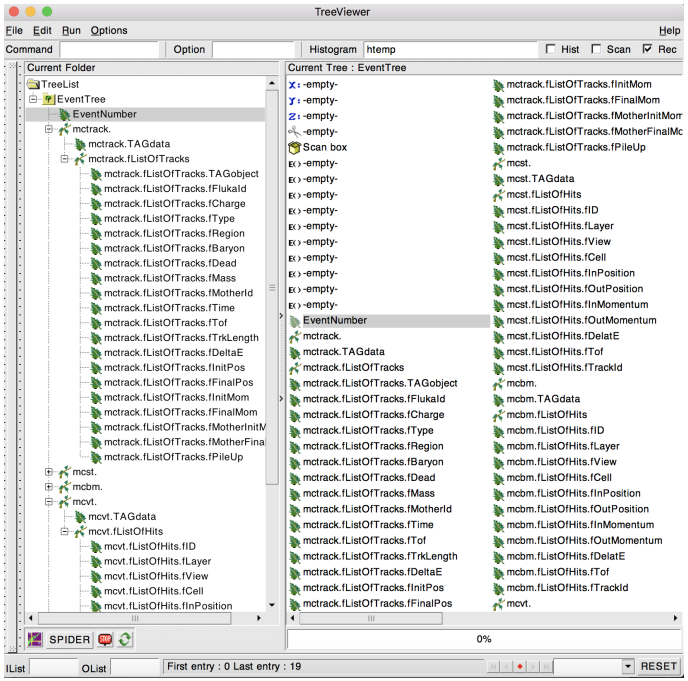
- Running **TAGsimulation** without option will automatically open Geant4 visualization (launched via macro vis.mac, located in G4simulation/macros/vis.mac)
- In Session menu, just type /run/beamOn n (with n number of incoming particles)...
- **NB :** Need of OpenGL to run visualization !



Geant4 output

- Geant4 in SHOE creates an output rootfile containing a TTree (EventTree).
- Two types of output : one is FLUKA compliant (Evento type), one fills the TTree with TClonesArray (see TAMCevent in TAMCbase).





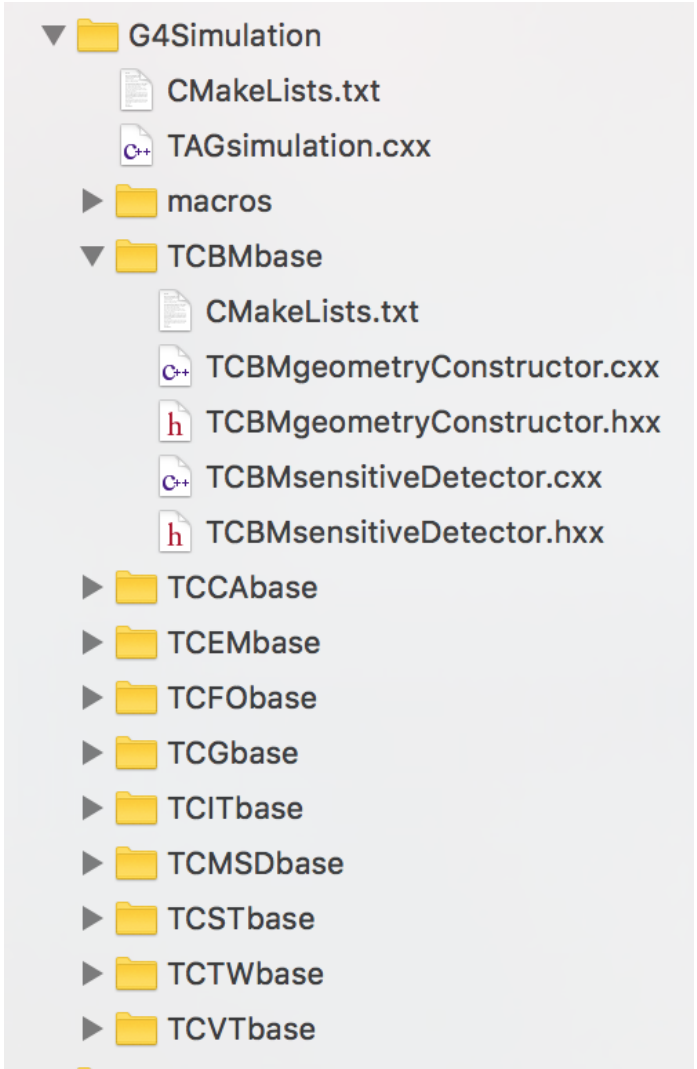
Geant4 output

- Output type can be switch on/off in TAGsimulation

```
int main(int argc,char** argv)
{
   // Construct the default run manager
   //
   G4RunManager* runManager = new G4RunManager;

   // initialize root file name
   TString rootFileName("ionO16.root");

   // initialise seed
   UInt_t seed = 0;

   // batch mode flag
   G4bool batchMode(false);

   // initialise physics list
   TString physListName("BIC");

   // select the output type (Evento tree or TAMCevent tree)
    G4bool kEvento(1);
```

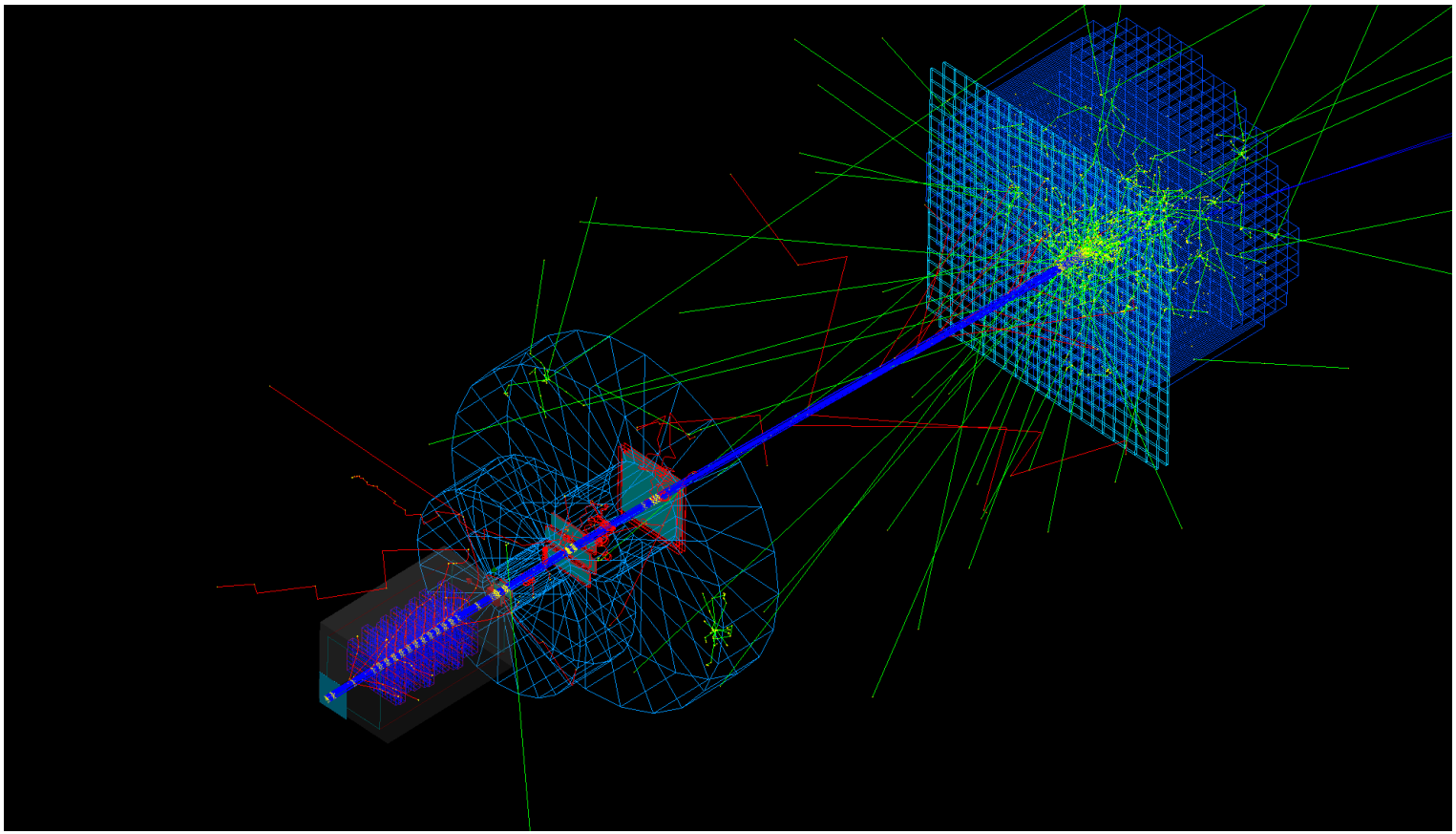
FOOT geometry

- For now, all detectors are implemented in Geant4 (including the magnets), but details are still missing in the IT and the MSD (e.g., passive parts).
- Each detector has its own subdirectory TC*base (similar to reconstruction code).
- To switch on/off a detector, simply change the FootGlobal.par config file.
- All detectors dimensions/distances/ materials are defined directly from TA*detector.map
- Target and beam definitions are defined from TAGdetector.map.



FOOT geometry

- Example of FOOT simulation with ^{16}O beam of 400 MeV/u on carbon target (NB : neutrons were removed from visualization)



Geant4 physics

- Different pre-defined physics lists (using different hadronic models) can be used in Geant4 :
  - BIC (Binary Cascade model),
  - INCL (Intranuclear Cascade model),
  - Bertini cascade model,
  - QMD (Quantum Molecular Dynamics)...
- **NB** : cuts in Geant4 are in range. For now, e-, e+ and gamma cuts have been set high (1 m) to avoid too important output files.

```
   // Physics list
#if G4VERSION_NUMBER < 1000
   printf("Geant4 v9 not supported");
   exit(0);
#else
    G4VModularPhysicsList* physics = 0x0;
   physListName.ToLower();
   if (physListName.Contains("bert"))
      physics = new QGSP_BERT();
   else if (physListName.Contains("bic"))
      physics = new QGSP_BIC();
   else if (physListName.Contains("incl"))
      physics = new QGSP_INCLXX();
    else if (physListName.Contains("qmd"))
       physics = new TCGphysicsQMD();
   else
      printf("\n\n No physics list defined !!\n\n"
#endif

   physics->SetVerboseLevel(0);
   physics->SetCutValue(1.0*m,"e-");
   physics->SetCutValue(1.0*m,"e+");
   physics->SetCutValue(1.0*m,"gamma");
   runManager->SetUserInitialization(physics);
```

Conclusion

Conclusion

- Geant4 is ready to be used in SHOE, but still work to do with MSD and IT (and magnets ?).
- Do not hesitate to go and see, give a try, and check if everything is correct (especially the geometry) :-)