Gate 6.2 - Fluorescence, Auger and PIXE

F. Smekens

09 july 2013

Fluorescence, auger and PIXE (Particle Induced X-ray Emission) phenomenons are making part of the global atomic deexcitation process. Although this one can be included in a pure Geant4 simulation, there is currently no way to use it easily in Gate v6.2. This documentation resumes the developments resolving this issue as well as some mechanisms of Geant4 useful to understand how to deal with atomic deexcitation.

1 Quick use

1.1 Macro

There is two solutions to include the atomic deexcitation in a simulation. The first one is the use of a predefined physics list (PL) that already includes the atomic deexcitation (this is true for most of predefined PL) :

- Electromagnetic (EM) physic only : e.g. "emlivermore"
- Hadronic physic (with EM physic extension) : e.g. "FTFP_BERT_LIV"

The complete list of PL is presented in the table 1. Note that we can not garanty the inherent presence of atomic deexcitation in all hadronic PL.

In order to add a predefined PL in the macro file, a single command can be applied. Note that, by using this solution, there is no possibility to change or add a particular process :

```
/gate/physics/addPhysicsList emlivermore
```

The second solution considers the case where the user don't know if the atomic deexcitation is available in a predefined PL or wants to choose manually the processes. The following command activate the atomic deexcitation :

```
/gate/physics/addAtomDeexcitation
```

Note that in both cases, fluorescence, auger and PIXE phenomenons are activated for all the regions of the simulation. Additional commands (already included in Geant4) let the possibility to deactivate individually the processes :

TABLE 1 – Predefined physics list for hadronic and EM physic.

Hadronic PL
FTFP_BERT, FTFP_BERT_TRV, FTF_BIC, LBE, LHEP, QBBC, QGSC_BERT, QGSP, QGSP_BERT, QGSP_BERT_CHIPS, QGSP_BERT_HP, QGSP_BIC, QGSP_BIC_HP, QGSP_FTFP_BERT, QGS_BIC, QGSP_INCLXX, QGSP_INCL_ABLA
EM PL in Gate (corresponding extensions for hadronic physic)
emstandard, emstandard_opt1(_EMV), emstandard_opt2(_EMX), emstandard_opt3(_EMY), emlivermore(_LIV), emlivermore_polar, empenelope(_PEN), emDNAphysics

```
/process/em/fluo  false    -> deactivate all
/process/em/auger  false   -> deactivate auger electrons
/process/em/pixe  false    -> deactivate PIXE
```

1.2 Example

An example for testing the presence of fluorescent photons can be found in the Gate repository. The simulation consists in generating mono-energetic photons inside a cube composed of a fluorescent medium (iodine by default). The activation of atomic deexcitation can be confirm using the physic verbose (Fig. 1) or by checking directly the presence of fluorescent photons with a energy spectrum actor (Fig 2).

2 Key points of the atomic deexcitation

2.1 Geant4 organization

G4VAtomDeexcitation Atomic deexcitation is process called by other processes. For example, fluorescence photon can be produced after a photoelectric effect or a compton scattering. That's why this process is not stored in the global physics list but in the loss table manager (*G4LossTableManager*), a class reachable by all others. However, this class is just a container and atomic deexcitation have to be instantiated manually as follows :

```
// Deexcitation
//
G4VAtomDeexcitation* de = new G4UAtomicDeexcitation();
G4LossTableManager::Instance()->SetAtomDeexcitation(de);
```

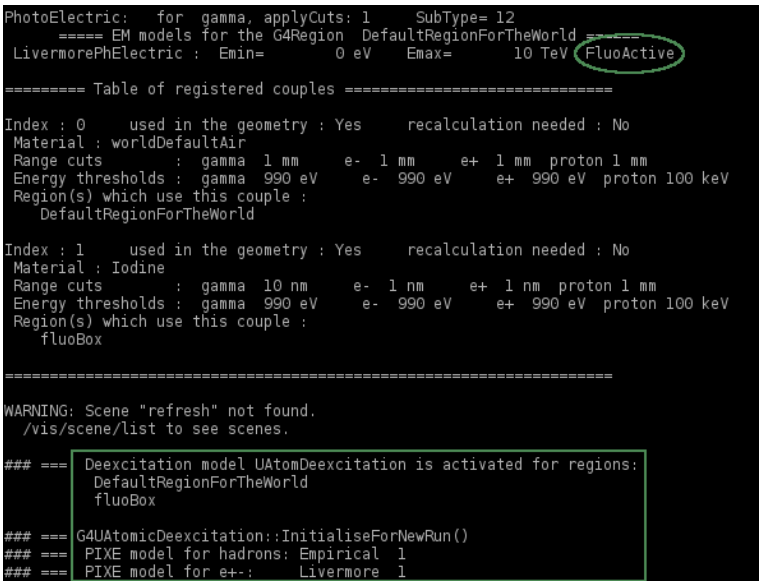

FIGURE 1 – Check atomic deexcitation activation using the physic verbose.

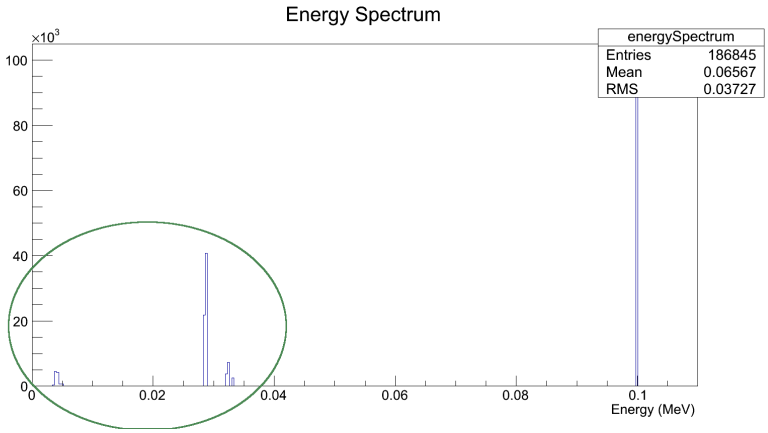

FIGURE 2 – Check atomic deexcitation activation using the energy spectrum actor and a iodine box.

During the construction phase, a process that effectively needs the atomic deexcitation will recover it in the loss table manager. In addition, the deexcitation has to be initialized. It appears that this initialization is effective when a *G4VEnergyLossProcess* process type (e.g. electron ionisation or multi scattering processes).

Activation flags The activation of atomic deexcitation is effective when the following conditions are fulfilled :

1. atomic deexcitation is instantiated,
2. atomic deexcitation is initialized,
3. atomic deexcitation is activated ("fluo flag"),
4. the region where the interaction occurs has been registered.

Note that, for each region, Geant4 gives the possibility to the user to activate individually the fluorescence, auger and PIXE processes.

2.2 Implementation in *GateVSource*

Atomic deexcitation instance As explain in the section 1.1, a special command has been added to instantiate the atomic deexcitation process. This command ensure that the process exists and that all modules are activated :

```
void GatePhysicsList::AddAtomDeexcitation()
{
  if(G4LossTableManager::Instance()->AtomDeexcitation() == NULL)
  {
    G4VAtomDeexcitation* de = new G4UAtomicDeexcitation();
    G4LossTableManager::Instance()->SetAtomDeexcitation(de);
  }
  opt->SetFluo(true);
  opt->SetAuger(true);
  opt->SetPIXE(true);
}
```

Region management In the current version of atomic deexcitation in Gate, all regions are automatically activated (fluorescence, auger, PIXE). For now, there is no way to deactivate a particular region. The only way is to modify the following function :

```
void GatePhysicsList::SetEmProcessOptions()
{
  ...
  // - register all regions in deexcitation process
  GateObjectStore *store = GateObjectStore::GetInstance();
  for(GateObjectStore::iterator it=store->begin() ; it!=store->end() ; ++it){
    opt->SetDeexcitationActiveRegion(it->first,true,true,true);
  }
}
```

3 Files and functions

- **GateRunManager** (source/general/...)
  `void GateRunManager::RunInitialization()`

- **GatePhysicsList** (source/physics/...)
  `void GatePhysicsList::ConstructPhysicsList(G4String)`
  `void GatePhysicsList::AddAtomDeexcitation()  (NEW)`
  `void GatePhysicsList::SetEmProcessOptions()`

- **GatePhysicsListMessenger** (source/physics/...)
  `"/gate/physics/AddAtomDeexcitation"  (NEW)`