Status and plans for de-excitation models

V. Ivanchenko

21 June 2017

Status of de-excitation module for 10.4beta

- Evaporation/FermiBreakUp/Photon evaporation are updated according to the plan for 10.4
  - Data structure for gamma levels is finelized G4PhotonEvaporation5.0
    - Twice less memory
    - Fixed several bug reports
  - Level energy and life time are double in the internal structure
    - G4float is used only for probabilities
  - Excited energy of a fragment is taken from the DB and not computed on-fly
  - All known bug reports are closed
  - Isomere production is working
- Correlated gamma decay
  - Not enabled by default
  - work for several isotopes (Co60) but provides crashes and very long loops if applied in general
- Updated GEM model not yet ready

Proposal for 10.4

- Move management of G4NuclearPolarization from RadioactiveDecay to the PhotonEvaporation model
  - PhotonEvaporation should fully control G4NuclearPolarization new/delete
  - Should fully decide if apply sampling of correlations or not
- The only one constrain:
  - G4RadioactiveDecay and G4ITDecay should have one and only one instance of G4PhotonEvaporation
    - no new/delete of G4PhotonEvaporation in run time
- For correlated gamma two possible regimes:
  - Fast and straightforward:
    - BreakUpChain(....)
      - Gamma cascade within G4PhotonEvaporation
      - only one G4NuclearPolarization object
    - May be applied from any type of consumer code
  - More problematic but still possible:
    - EmittedFragment(....)
      - polarization should be kept and transport of an isotope will be allowed
        - Correlation cannot be sampled if other radioactive decay chain is executed in between
      - The fragment property should be cached inside G4PhotonEvaporation
      - Applied only by flag of the radioactive decay

Proposal for 10.4: sampling of correlations

- G4PhotonEvaporation will have logic to setup flags
- G4Fragment will not carry out G4NuclearPolarization pointer anymore
- G4GammaTransition will be responsible for interaction with G4PolarizationTransition
  - will have logic to create/delete G4NuclearPolarization objects
  - Call G4Polarizationtransition
- Instead of 4 public interfaces G4PolarizationTransition will have only one:
  - SampleTransition(......)
  - All other methods will be private

CPU/memory monitoring

- Since 10.3 we have CPU/memory performance problems
- A.Dotti has discovered that the coalescence sub-model of the Bertini cascade was pratially responsible for the memory increase
- We are not fully confident if the source of remaining penalty is still Bertini or de-excitation module, or something else
- Possible checks to be performed:
  - Check CPU/memory for transition region Bertini/FTF 4 – 5 GeV
  - Introduce a new class G4PrecoDummyModel which may be used intead of the default preco model
    - If it is enabled, the default preco/de-excitation fully diabled
  - Introduce also G4DummyPhotonEvaporation