Gunter Folger

Geant4 Workshop Catania 2009

Integer A&Z

- Why change
  - Several bugs due to inconsistent rounding of mass number A
    - Resulting in energy non-conservation
    - less so for charge number Z
  - Readability:
    - G4int N = G4int(elm->GetN()+0.5) - iz;
      G4double A = G4double(G4int(elm->GetN()+0.5));
      A = G4double((*isv)[0]->GetN());

- When --- assuming we decide to change
  - After 9.3,
    - 9.3 very close
    - Changes in many classes needed
  - early 2010, to allow for testing

Charge, Neutron, and Mass Number are Integer

- Two main `sources' of A, Z, N in hadronics:
  - G4Element
    - Processes, not using G4HadronicProcess
  - Cannot change anything here
  - G4Nucleus
    - Models via GetN() and GetZ()
  - Suggest to migrate

How to migrate

- Top down approach
  - Change G4HadronicProcess (?), G4CrossSectionDataStore and G4Nucleus first
  - Adapt models/util classes one by one in second step
- I do not see this as being a choice

How to Migrate – Interface in G4Nucleus

Choice of Implementation

- Keep current access methods unchanged
  - G4Nucleus has
    - G4double GetZ() charge
    - G4double GetN() mass number

  Extend interface interface with new methods returning G4int
    - G4int GetZ_asInt() charge
    - G4int GetN_asInt() Number of neutrons
    - G4int GetA_asInt() mass number

- Re-use interface changing signature in G4Nucleus:
    - G4int GetZ()
    - G4int GetN()

Conclusion – as agreed

- We DO migrate
- We start in January
- Interface of G4Nucleus
  - ~~Re-use interface, or~~
  - Extend interface, new methods
    - G4int GetZ_asInt()
    - G4int GetN_asInt()
    - G4int GetA_asInt()
- Top down approach