

Using Standard G4 Features in Hadronics

Dennis Wright

Geant4 Collaboration Workshop

19 October 2009

Motivation

- Several hadronic models independently implement methods which are already available in Geant4/CLHEP
- This leads to maintenance and debugging problems
- Need to use standard features throughout hadronics code

Examples (1)

- Bertini, LEP, HEP
 - each use their own Lorentz boost algorithms
 - should use CLHEP methods
 - same for factorials, gamma functions, ...
- CHIPS, G4HadronCrossSections, LEP, HEP, Bertini
 - could use G4PhysicsVector in many places
- Models which are probably OK (because they originated within Geant4)
 - Binary cascade, QMD, HP neutrons, G4Precompound
 - but these need to be checked anyway

Examples (2)

- Some codes still use their own sets of units, then convert to G4 units at the end
- Some codes still use their own particle types, then convert to G4 types at the end
 - INCL/ABLA, Bertini
- Hadronic framework uses its own set of track status codes then converts to Geant4 codes when transferring the output
 - change this ?
- Anything else?

Plan

- Identify which migrations to standard G4 features to do
 - some will be easy
 - using CLHEP factorials, gamma functions, ...
 - internal units
 - some will be hard
 - Lorentz boost in LEP, HEP
- Identify who will do changes