Generalities on processes

Geant4 Short Course

3-5 November 2005

Bordeaux

Marc Verderi
Ecole Polytechnique - LLR

Introduction

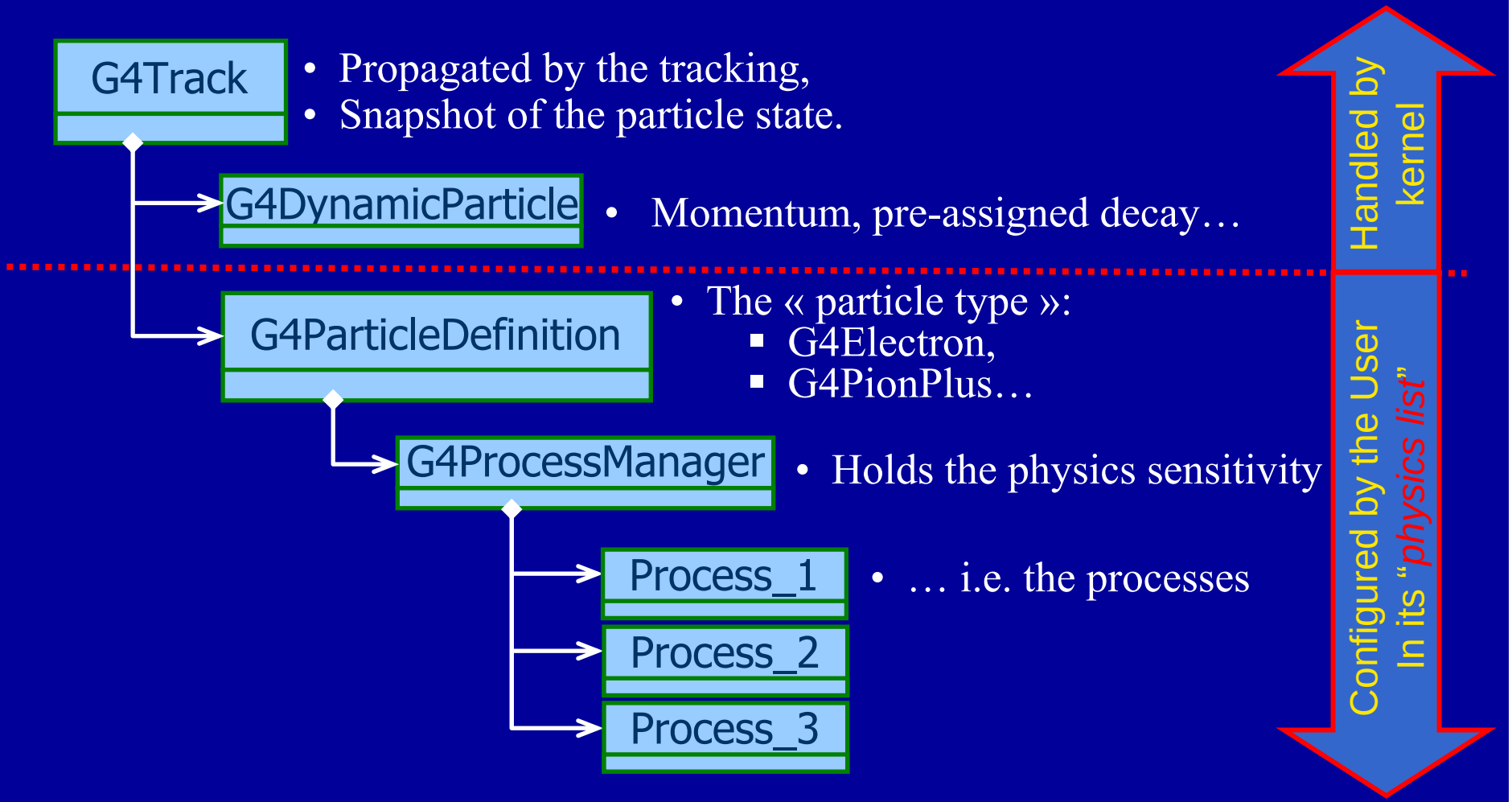
- Geant4, as a “virtual reality” software, has to “catch” reality through models.
- The model for processes is realized by an abstract interface, G4VProcess
  - Shared by all processes
  - Thanks to object orientation design
- Present here this abstract interface (presentations of concrete processes follow)
  - and present the related way of how processes are handled by the tracking
- As any modelization, limitations exist. They arise here with the need for setting “cuts”
  - they are “production cuts”, presented in the last section

Contents

I. From G4Track to processes
II. The process interface
   – G4VProcess
   – How processes are used by the stepping
III. The production cuts

I. From G4Track to processes

From G4Track to processes



II. The process interface

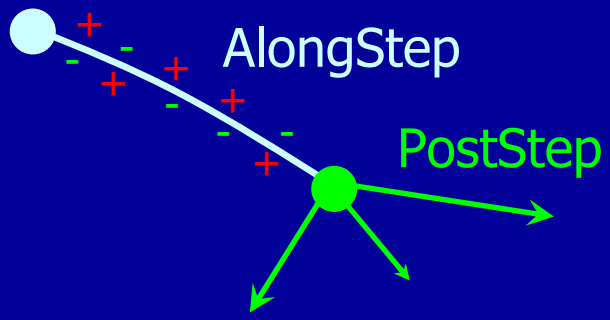
Speak about:

G4VProcess

The stepping

G4VProcess: 3 kind of actions (1)

- Abstract class defining the common interface of all processes in Geant4:
  - Used by all « physics » processes
  - but is also used by the transportation, etc…
  - Defined in source/processes/management
- Define three kinds of actions:
  - AtRest actions:
    - Decay, e^+ annihilation …
  - AlongStep actions:
    - To describe continuous (inter)actions, occuring along the path of the particle, like ionisation;
  - PostStep actions:
    - For describing point-like (inter)actions, like decay in flight, hard radiation…



G4VProcess: 3 kind of actions (2)

- A process can implement any combination of the three AtRest, AlongStep and PostStep actions:
  - eg: decay = AtRest + PostStep
- And from a practical point of view, a set on intermediate classes exist implementing various combinations of actions:
  - For example:
    - G4VDiscreteProcess: only PostStep actions;
    - G4VContinuousDiscreteProcess: AlongStep + PostStep actions;
    - ...
  - In case you foresee to implement your own processes.

G4VProcess: action methods

- Each action defines two methods:
  - GetPhysicalInteractionLength():
    - Used to ***limit the step size***:
      - either because the process « triggers » an interaction, a decay
      - or any other reasons, like fraction of energy loss, geometry boundary, user's limit …
  - DoIt():
    - Implements the ***actual action*** to be applied on the track;
    - And the related production of secondaries.

G4VProcess : actions summary

- The « action » methods are thus:
  - AtRestGetPhysicalInteractionLength(), AtRestDoIt();
  - AlongStepGetPhysicalInteractionLength(), AlongStepDoIt();
  - PostStepGetPhysicalInteractionLength(), PostStepDoIt();
- G4VProcess defines other methods:
  - G4bool IsApplicable(const G4ParticleDefinition &);
    - which returns « true » if the process is applicable to the given particle type
  - And methods called at the beginning and end of tracking of a particle, etc…

How the Stepping handles processes

- The stepping treats processes ***generically***:
  - The stepping does not know(*) what processes it is handling;
- The stepping makes the processes to:
  - Cooperate for AlongStep actions;
  - Compete for PostStep and AtRest actions;
- Particular treatments are also possible on process request, which can ask to be
  - forced:
    - PostStepDoIt() action applied anyway;
      - e.g. transportation to update G4Track geom. info
  - conditionallyForced:
    - PostStepDoIt() applied if AlongStep has limited the step;
  - etc …

(*) almost: some exception for transportation

Stepping Invokation Sequence of Processes for a particle travelling

1. At the beginning of the step, determine the step length:
   - Consider all processes attached to the current G4Track;
   - Define the step length as the smallest of the lengths among:
     - All AlongStepGetPhysicalInteractionLenght()
     - All PostStepGetPhysicalInteractionLength()
2. Apply ***all*** AlongStepDoIt() actions, « at once »:
   - Changes computed from particle state at the beginning of the step;
   - Accumulated in the G4Step;
   - Then applied to the G4Track, from the G4Step.
3. Apply PostStepDoIt() action(s) « sequentially », as long as the particle is alive:
   - Apply PostStepDoIt() of process which proposed the smallest step length;
   - And apply « forced » and « conditionnally forced » actions.

Stepping Invokation Sequence of Processes for a Particle at Rest

1. If the particle is at rest, *is stable and can't annihilate*, it is killed by the tracking:
   - More properly said: if a particle at rest has no « AtRest » actions defined, it is killed.
2. Otherwise determine the lifetime:
   - Take the smallest time among:
     - All AtRestGetPhysicalInteractionLenght()
       - Called « physical interation lenght » but returns a time;
3. Apply AtRestDoIt() action of process which returned the smallest time.

G4VProcess & G4ProcessManager

- G4ProcessManager maintains three vectors of actions:
  - One for the AtRest methods of the particle;
  - One for the AlongStep ones;
  - And one for the PostStep actions.
- These are these vectors the user sets up in the "physics list" and which are used by the tracking.
- Note that the process ordering provided by/to the G4ProcessManager vectors ***IS*** relevant.

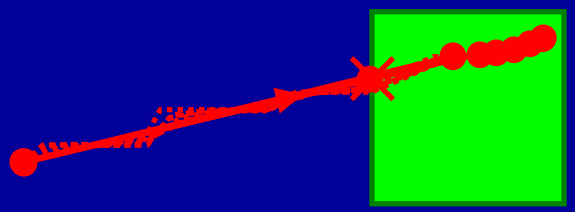
A word about processes ordering

- **The ordering of processes matters !**
- Ordering of following processes is critical for a few of them:
  - Assuming n processes, the ordering of the AlongGetPhysicalInteractionLength() of the last processes should be:

    [n-2] …

    [n-1] multiple scattering

    [n] transportation
- Why ?
  - Processes return a « true path length »;
  - The multiple scattering « virtually folds up » this true path length into a ***shorter*** « geometrical » path length;
  - Based on this new length, the transportation can geometrically limits the step.



- Other processes ordering usually does not matter.

III. The production cuts;

Speak about:

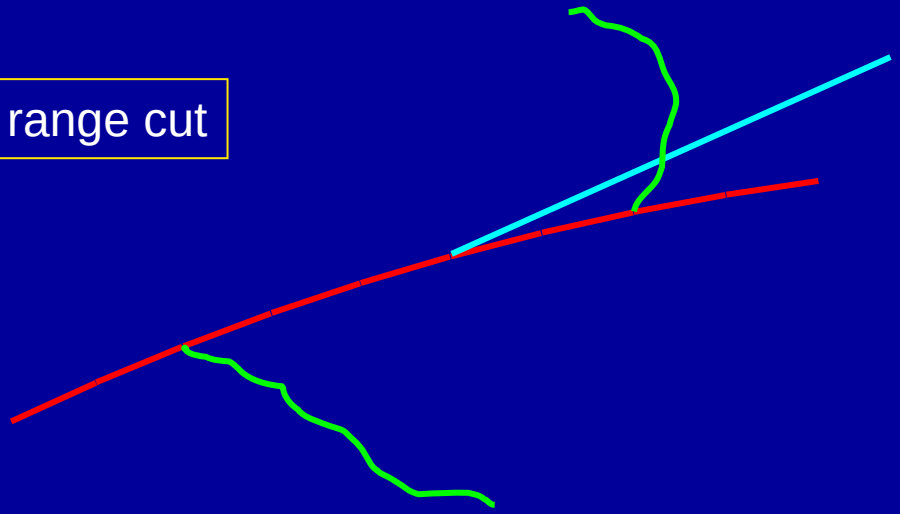
- Why production cuts are needed
- The cuts scheme in Geant4

The “cuts” in Geant4

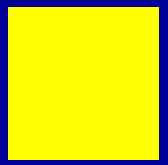
- In Geant4 there is no tracking cut:
  - Particles are tracked down to a zero range/kinetic energy;
- Only production cuts exist;
  - ie cuts allowing a particle to born or not;
- Why production cuts are needed ?
- Some electromagnetic processes involve infrared divergences:
  - This leads to an infinity[huge number] of smaller and smaller energy photons[electrons] (like in bremstrahlung, δ-ray productions);
  - Production cuts limit this production to particles above the threshold;
  - The remaining, divergent part (divergent in # particle, not in energy…) is treated as a « net » continuous effect (ie « AlongStep » action);
- Production cuts define the boundary between discrete and continuous energy loss for these processes
  - These cuts have to be customized by the user for its application.

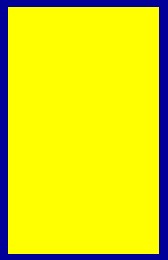
Low range cut

High range cut



Local energy deposit





Cut scheme in Geant4

- Mind that in Geant4 « cuts » are expressed in range
  - Processes do use an energy cut, but this range to energy cut conversion is provided by Geant4 for each particle type and each material
- Geant4 used to allow a unique range cut for the entire simulation
  - Rigourous from a consistency point of view, but felt as to rigid
- Now, the G4Region concept has been introduced to allow to define one range cut per region
  - e.g. hadronic calorimeters and silicon vertex detector are not forced to deal with a common range cut anymore
- To tell the truth, it is possible, though not much recommended, to specify a range cut per particle type (and per region).

Conclusion/summary

- All processes share the same interface, G4VProcess:
  - This allows Geant4 to treat processes generically:
  - Three types of actions are defined:
    - AtRest (compete), AlongStep (cooperate), PostStep (compete)
    - Each action define a “GetPhysicalInterationLenght()” and a “DoIt()” method
- Processes are attached to the particle by its G4ProcessManager
  - This is the way the particle acquires its sensitivity to physics
  - This G4ProcessManager is set up in the “physics list”
    - Please be careful of the multiple scattering and transportation ordering
- Some processes require “cuts”, i.e. “production threshold”:
  - to be defined to absorb infrared divergences into a continuous energy loss contribution
  - That needs to be tuned by the user for its particular application
- One range cut can be specified per region