Physics II.: Physics Process

Mihaly Novak (CERN, EP-SFT)

Geant4 Tutorial at Jefferson Lab, 25 March 2024

Geant4.11.2.p01

OUTLINE

- **Overview of Geant4 physics components**
- **Some relevant key concepts:** `G4Track`, `G4Step`, etc...
- **The Geant4 physics process interface(es)**

PHYSICS COMPONENTS

- Overview of Geant4 physics components
- Some relevant key concepts: `G4Track`, `G4Step`, etc…
- The Geant4 physics process interface(es)

Physics Components

- **Geant4 provides a vide variety of physics components**
- **The building blocks of these components are *Processes*es:**
  - a process describes a well defined interaction of (a) particle(s) with matter
  - describe = determines *when* the interaction happens and what the *result is*
  - processes provide these information through a `G4VProcess` interface (later)
  - Geant4 provides a huge number of such processes
  - users might introduce their own process(es) easily by implementing the general process interface
- **Processes are classified as:**
  - Electromagnetic
  - Hadronic
  - Decay
  - Parameterized
  - Transportation

Physics Components

- **Geant4 Physics: Electromagnetic**
  - *the standard EM part*: provides a complete set of EM interactions (processes) of charged particles and gammas from 1 keV to ~PeV
  - *the low energy EM part*: includes special treatments for low energy e-/+, gammas and charged hadrons:
    - more sophisticated approximations valid down to lower energies e.g. more atomic shell structure details
    - some of these processes will be valid down to below keV but some can be used only up to few GeV
  - *optical photons*: interactions special only for long wavelength photons
    - processes for reflection/refraction, absorption, wavelength shifting, (special) Rayleigh scattering

Physics Components

- **Geant4 Physics: Hadronic**
  - pure hadronic interactions for 0 to ~TeV
    - elastic, inelastic, capture, fission
  - radioactive decay:
    - both at-rest and in-flight
  - photo-nuclear interaction from ~10 MeV up to ~TeV
  - lepto-nuclear interaction from ~10 MeV up to ~TeV
    - e+ and e- induced nuclear reactions
    - muon induced nuclear reactions

Physics Components

- **Geant4 Physics: Decay, Parameterized and Transportation**
  - decay processes includes:
    - weak decay (leptonic, semi-leptonic decay, radioactive decay of nuclei)
    - electromagnetic decay (π^0, Σ^0, etc.)
    - strong decay not included here (they are part of hadronic models)
  - parameterized process:
    - EM shower generation based on parameters obtained from averaged events
    - used as fast simulation in case of complex detectors: fast but less accurate
  - transportation process:
    - special process that responsible to propagate the particles through the geometry
    - need to be assigned to each particle

Some relevant key concepts

- Overview of Geant4 physics components
- **Some relevant key concepts:** `G4Track`, `G4Step`, etc…
- The Geant4 physics process interface(es)

Some relevant key concepts

- **Geant4 propagates `G4Track` objects in a `G4Step`-by-`G4Step` way**

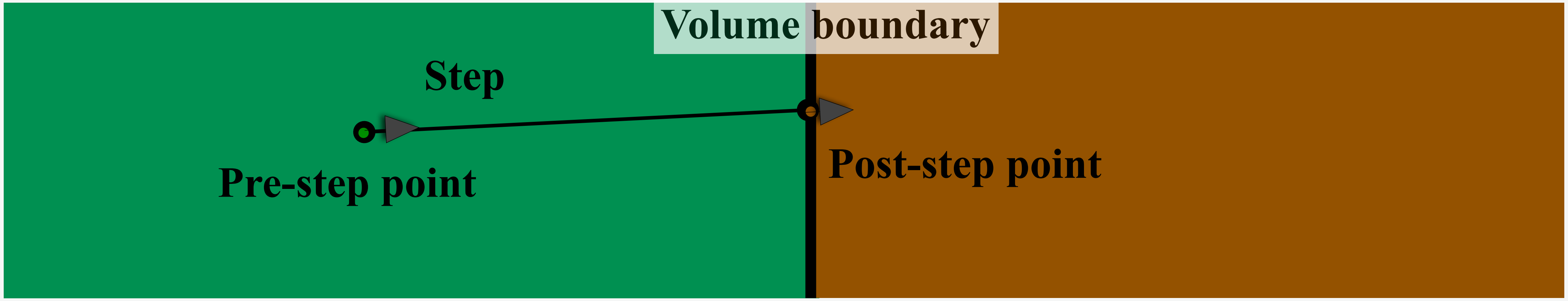
- **`G4Track`:**
  - a **`G4Track`** object represents/describes **the state of a particle** that is under simulation in a given instant of the time (i.e. **a** given **time point**)
  - a snapshot of a particle **without** keeping any **information regarding the past**

  - its **`G4ParticleDefinition`** stores **static** particle **properties** (charge, mass, etc.) as it describes a particle type (e.g. **`G4Electron`**)
  - its **`G4DynamicParticle`** stores **dynamic** particle **properties** (energy, momentum, etc.)
  - while all **G4Track**-s, describing the same particle type, share the same, unique **`G4ParticleDefinition`** object of the given type (e.g. **`G4Electron`**), each individual track has its own **`G4DynamicParticle`** object
  - e.g. electrons: only one **`G4Electron`** object but as many **`G4DynamicParticle`** as electron **G4Track**-s

  - the **`G4Track`** object is propagated in a ***step-by-step*** way during the simulation of a given particle: the dynamic properties are continuously updated to reflect the current state
  - continuously updated: even within one simulation step
  - step? step-by-step? what about the difference between two such states within a step?

Some relevant key concepts

- **Geant4 propagates `G4Track` objects in a `G4Step`-by-`G4Step` way**
  - **`G4Step`:**
    - a **`G4Step`** object can provide the information regarding the **change in** the **state of the particle (**that is under tracking) **within a** simulation **step** (i.e. **delta**)
    - has two **`G4StepPoint`**-s, pre- and post-step points, that stores information (position, direction, energy, material, volume, etc…) that belong to the corresponding point (space/time/step)
    - these are updated in a step-by-step way: the post-step point of the previous step becomes the pre-step point of the next step (when the next step starts)
    - **(important)** if a step is limited by the geometry (i.e. by a volume boundary), the post-step point:
      - **physically** stands **on the boundary** (the step status of the post step point i.e. `G4Step::GetPostStepPoint()->GetStepStatus()` is `fGeomBoundary`)
      - **logically** belongs **to the next volume**
      - since these "*boundary*" **`G4Step`**-s have information both regarding the previous and the next volumes/materials, boundary processes (e.g. reflection, refractions and transition radiation) can be simulated



Some relevant key concepts

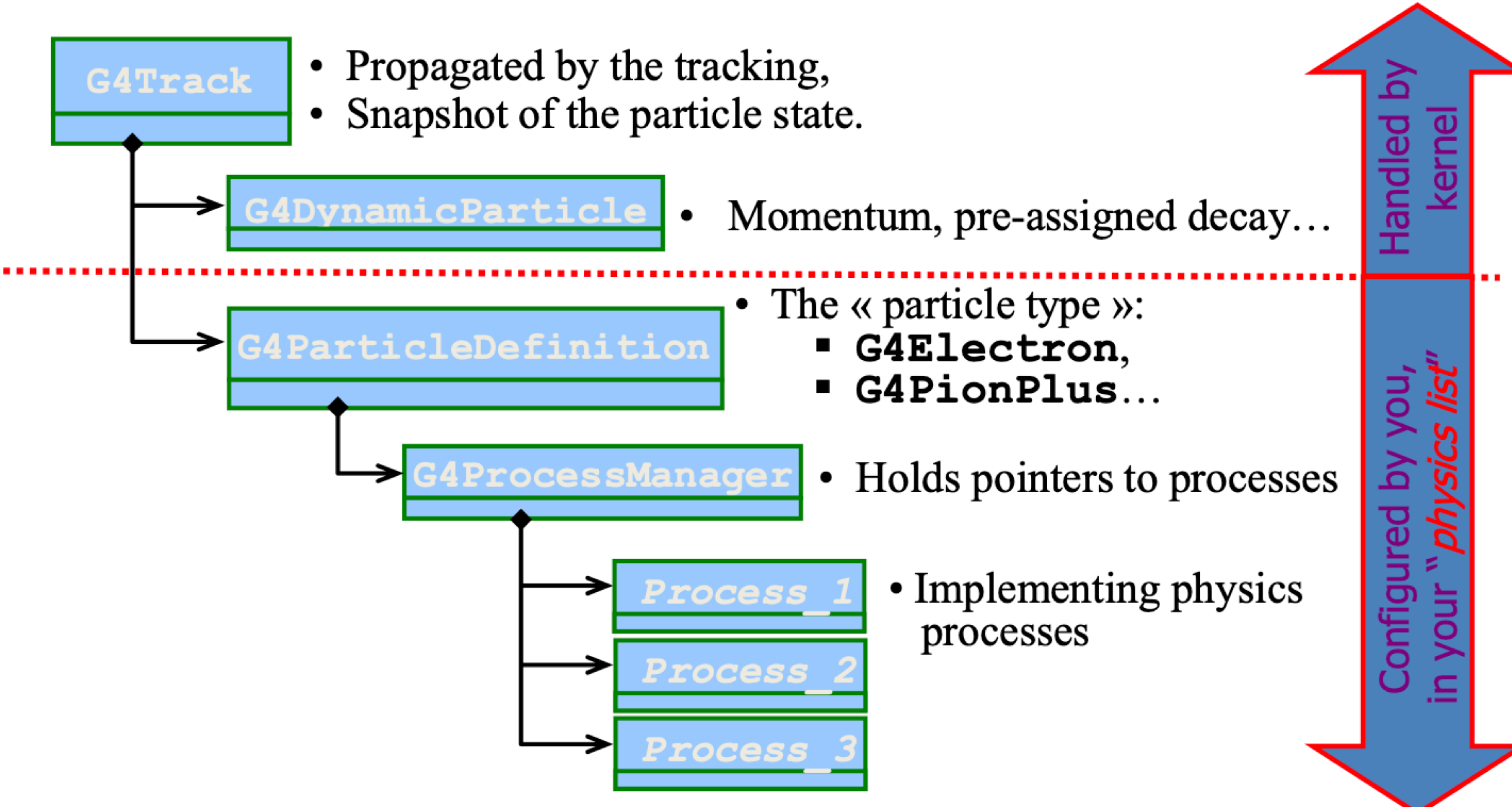
Geant4 propagates `G4Track` objects in a `G4Step`-by-`G4Step` way

- **`G4Step`:**
  - a **`G4Step`** object can provide the information regarding the **change in** the **state of the particle (**that is under tracking) **within a** simulation **step** (i.e. **delta**)
  - has two **`G4StepPoint`**-s, pre- and post-step points, that stores information (position, direction, energy, material, volume, etc…) that belong to the corresponding point (space/time/step)
  - these are updated in a step-by-step way: the post-step point of the previous step becomes the pre-step point of the next step (when the next step starts)
  - **(important)** if a step is limited by the geometry (i.e. by a volume boundary), the post-step point:
    - **physically** stands **on the boundary** (the step status of the post step point i.e. `G4Step::GetPostStepPoint()->GetStepStatus()` is `fGeomBoundary`)
    - **logically** belongs **to the next volume**
    - since these "*boundary*" **`G4Step`**-s have information both regarding the previous and the next volumes/materials, boundary processes (e.g. reflection, refractions and transition radiation) can be simulated
  - the **`G4Track`** object, that is under tracking i.e. generates information for the **`G4Step`** object, can be obtained from the step by the **`G4Step::GetTrack()`** method and the other way around **`G4Track::GetStep()`**

Some relevant key concepts

- **Geant4 propagates** `G4Track` **objects in a** `G4Step`**-by-**`G4Step` **way**
- the actual **details of a** simulation **step** (its computation, nature) **are determined by the** particle (type, kinematics, etc.) and its **possible interactions**
- **<u>Process</u>:** the Geant4 concept (with the `G4VProcess` interface) for describing interactions
- the **possible interactions depend** (primarily) **on the particle type**
- **the list** of possible interactions of a given particle type **is declared in the <u>Physics List</u>**
- this list is **stored in a** `G4ProcessManager` object**:**
  - each `G4ParticleDefinition` object (particle type) has a process manager
  - that holds a list of `G4VProcess` objects that has been assigned to the particle
  - when simulating a `G4Track`, with a given type of particle, the corresponding `G4ParticleDefinition` is obtained from the `G4Track` then the `G4ProcessManager`
  - the `G4ProcessManager` provides then the list of `G4VProcess`-es that are used to compute the step

Some relevant key concepts



PHYSICS COMPONENTS

- Overview of Geant4 physics components
- Some relevant key concepts: `G4Track`, `G4Step`, etc…
- **The Geant4 physics process interface(es)**

Physics Process Interface

- **The** `G4VProcess` **is:**
  - the ***general*** Geant4 physics process ***interface*** for describing any interactions
  - at each step, each interaction must provide information such as:
    - How far(space/time) this particle goes till the next interaction of the given type ?
    - What happens in the interaction ? (post interaction primary state + secondaries)
  - `G4VProcess` provides *interface methods* for this information flow:
    - `GetPhysicalInteractionLength()` - to provide the interaction length
    - `DoIt()` - to perform the transformation from the pre- to the post-interaction state
  - **NOTE:** the **step calculation is the same** for all type of particle! Excellent design that abstracts away all the differences that are due to the particle type
  - in general, the particle can interact with matter:
    - AlongStep - **continuously**, while moves from the pre- to the post-step point
    - PostStep - at the **discrete** post-step point of the step (*well-located in space*)
    - **AtRest** - when it stopes (*well-located in time*)
  - for each form of the above interactions, the process needs to implement both the corresponding `GetPhysicalInteractionLength()` and `DoIt()` methods
  - a process might be the combination of some or all of the above(6 methods)

Physics Process Interface: example processes

- **Discrete process: Compton scattering**
  - length of the step to the interaction determined by the cross section and the interaction happens at the post-step point
    - PostStepGetPhysicalInteractionLength() and PostStepDoIt()
  - a `G4VDiscreteProcess` derived from the generic `G4VProcess` interface
- **Continuous process: Cherenkov effect**
  - photons are created along the step (# proportional to the step length)
    - AlongStepGetPhysicalInteractionLength() and AlongStepDoIt()
  - a `G4VContinuousProcess` derived from the generic `G4VProcess` interface
- **At-Rest process: muon minus capture at rest**
  - muon has already stopped (zero kinetic energy) so time is the relevant
    - AtRestGetPhysicalInteractionLength() and AtRestDoIt()
  - a `G4VAtRestProcess` derived from the generic `G4VProcess` interface

Physics Process Interface: example processes

- **Continuous + Discrete process: bremsstrahlung (ionization)**
  - low energy photons (electrons) are not generated, the corresponding energy loss is deposited along the step as **continuous** process
  - energetic photons (electrons) are generated in **discrete** interaction
  - secondary photon (electron) production threshold separates the two continuous and discrete parts (see later)
    - `PostStepGetPhysicalInteractionLength()` and `PostStepDoIt()`
    - `AlongStepGetPhysicalInteractionLength()` and `AlongStepDoIt()`
  - a `G4VContinuousDiscreteProcess` derived from the generic `G4VProcess` interface

- **Discrete + At-Rest process: positron annihilation**
  - in-flight annihilation as a **discrete** process, determined by the cross section
  - **at-rest** annihilation, when the positron has already stopped
    - `PostStepGetPhysicalInteractionLength()` and `PostStepDoIt()`
    - `AtRestGetPhysicalInteractionLength()` and `AtRestDoIt()`
  - a `G4VRestDiscreteProcess` derived from the generic `G4VProcess` interface

Physics Process Interface: process management

- **at initialisation:**
  - many processes (i.e. possible interactions) might be assigned to a given particle type, e.g. gamma photon:
    - *particle* type i.e. `G4ParticleDefinition`: `G4Gamma`
    - *processes:* e+/e- *pair-production*, *Compton* and *Rayleigh* scat., *photoelectric* effect, etc.
  - the *processes* (all implements the `G4VProcess` interface) are assigned to the corresponding `G4ParticleDefinition` in the Physics List (EM constructor)
  - each `G4ParticleDefinition` stores the list of assigned *processes* in its `G4ProcessManager`

```
void YourPhysicsList::ConstructEM() {
  // get the physics list helper
  // it will be used to assign processes to particles
  G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();
  auto particleIterator   = GetParticleIterator();
  particleIterator->reset();
  // iterate over the list of particles constructed in ConstructParticle()
  while( (*particleIterator)() ) {
    // get the current particle definition
    G4ParticleDefinition* particleDef   = particleIterator->value();
    // if the current particle is the appropriate one => add EM processes
    if ( particleDef == G4Gamma::Definition() ) {
      // add physics processes to gamma particle here
      ph->RegisterProcess(new G4GammaConversion(),  particleDef);
      ...
      ...
    } else if ( particleDef == G4Electron::Definition() ) {
      // add physics processes to electron here
      ph->RegisterProcess(new G4eBremsstrahlung(), particleDef);
```

Physics Process Interface: process management

- **at initialisation:**
  - many processes (i.e. possible interactions) might be assigned to a given particle type, e.g. gamma photon:
    - *particle* type i.e. `G4ParticleDefinition`: `G4Gamma`
    - *processes:* e+/e- *pair-production*, *Compton* and *Rayleigh* scat., *photoelectric* effect, etc.
  - the *processes* (all implements the `G4VProcess` interface) are assigned to the corresponding `G4ParticleDefinition` in the Physics List (EM constructor)
  - each `G4ParticleDefinition` stores the list of assigned *processes* in its `G4ProcessManager`

```
phot:   for  gamma    SubType= 12  BuildTable= 0
      LambdaPrime table from 200 keV to 100 TeV in 61 bins
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
 LivermorePhElectric :  Emin=        0 eV     Emax=      100 TeV   AngularGenSauterGavrila  FluoActive

compt:   for  gamma    SubType= 13  BuildTable= 1
      Lambda table from 100 eV  to 1 MeV, 7 bins per decade, spline: 1
      LambdaPrime table from 1 MeV to 100 TeV in 56 bins
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
        Klein-Nishina :  Emin=        0 eV     Emax=      100 TeV

conv:   for  gamma    SubType= 14  BuildTable= 1
      Lambda table from 1.022 MeV to 100 TeV, 18 bins per decade, spline: 1
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
        BetheHeitler :  Emin=        0 eV     Emax=       80 GeV   AngularGenUrban
     BetheHeitlerLPM :  Emin=       80 GeV    Emax=      100 TeV   AngularGenUrban

Rayl:   for  gamma    SubType= 11  BuildTable= 1
      Lambda table from 100 eV  to 100 keV, 7 bins per decade, spline: 0
      LambdaPrime table from 100 keV to 100 TeV in 63 bins
      ===== EM models for the G4Region  DefaultRegionForTheWorld ======
   LivermoreRayleigh :  Emin=        0 eV     Emax=      100 TeV   CullenGenerator
```

Physics Process Interface: process management

- **at run-time:** (caveat: rather simplified description of a step)
  - when simulating a `G4Track`, with a given type of particle
  - the corresponding `G4ParticleDefinition` is obtained from the `G4Track`
  - then its `G4ProcessManager`, that provides the list of `G4VProcess`-es
    - list of discrete, or continuous or at-rest processes assigned to the particle that are used to calculate the simulation step
  - all processes follow the `G4VProcess` process interface:
    - each implements their interaction-length and do-it interface method(s)
  - at the pre-step point, each processes assigned to the particle:
    - will be asked to provide its physics-interaction length
    - transportation will also provide its length i.e. geometry related constraints
    - the shortest among these length will be selected as the current step length
    - it determines the post-step point (without field in case of charged particles)
    - it determines the interaction (i.e. process) that happens (if any) at that point
  - the track will be transported to the post step point:
    - the `DoIt()` process interface method(s) will be invoked to perform the interaction(s)
    - `AlongStepDoIt()` **all** processes in **each steps**: describe **continuous** interactions
    - `PostStepDoIt()` **at most one**: describe **discrete** interactions that compete (also with the continuous interactions: invoked only if the given discrete process limited the step)

Physics Process

Some of the important special processes will be discussed in the Electromagnetic and Hadronic physics lectures