Basic Structure of the Geant4 Simulation Toolkit

http://cern.ch/geant4

The full set of lecture notes of this Geant4 Course is available at
http://www.ge.infn.it/geant4/events/nss2004/geant4course.html

Contents

- Geant4 categories and kernel
  - major functions
  - organization and architecture
- Tracking and processes
  - how tracking is done
  - overview of processes
- User classes
- System of units, intercoms, environment variables

Geant4 Kernel

- ***Provides central functionality of the toolkit***
  - ***handles runs, events, tracks, steps, hits, trajectories***
  - ***implements Geant4 as a state machine***
  - ***provides a framework for:***
    - ***physics processes***
    - ***visualization drivers***
    - ***GUIs***
    - ***persistency***
    - ***histogramming/analysis***
    - ***user code***

Run

- ***A run is a collection of events which are produced under identical conditions***
- ***Within a run, user cannot change:***
  - ***Detector or apparatus geometry***
  - ***Physics process settings***
- ***By analogy to high energy physics a Geant4 run begins with the command "BeamOn"***
  - ***Detector is inaccessible once beam is on***
- ***At beginning of run:***
  - Geometry is optimized for navigation
  - Cross sections are calculated according to materials in setup
  - Low-energy cutoff values are defined

Event

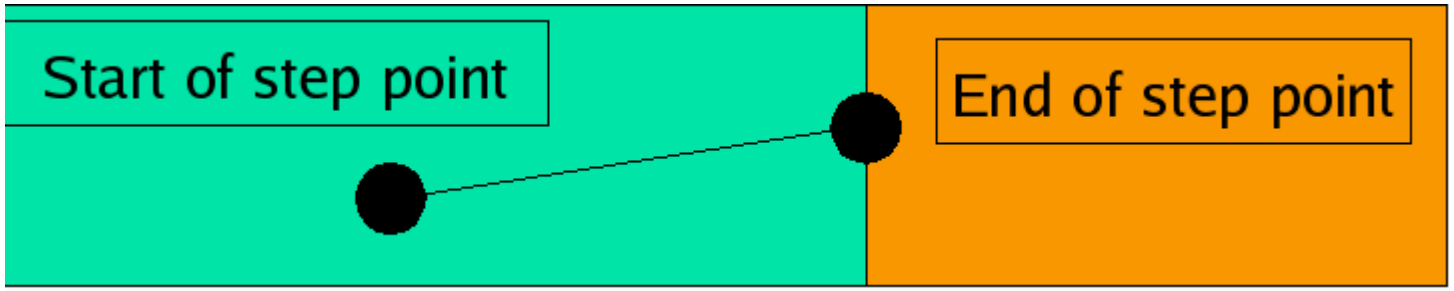
- At beginning of processing, an event contains primary particles (from generator, particle gun, ...), which are pushed onto a stack
- During processing, each particle is popped from the stack and tracked. When the stack is empty, processing of the event is over
- The class G4Event represents an event. At the end of processing it has the following objects:
  - List of primary vertices and particles (the input)
  - Hits collections
  - Trajectory collections (optional)
  - Digitizations collections (optional)

Track

- A track is a snapshot of a particle within its environment
  - as the particle moves, the quantities in the snapshot change
  - at any particular instance, a track has position, physical quantities
  - it is not a collection of steps
- Track object lifetime
  - created by a generator or physics process (e.g. decay)
  - deleted when it:
    - leaves world volume
    - disappears (particle decays or is absorbed)
    - goes to zero energy and no “at rest” process is defined
    - user kills it
- No track object survives the end of an event (not persistent)
  - User must take action to store track record in trajectory

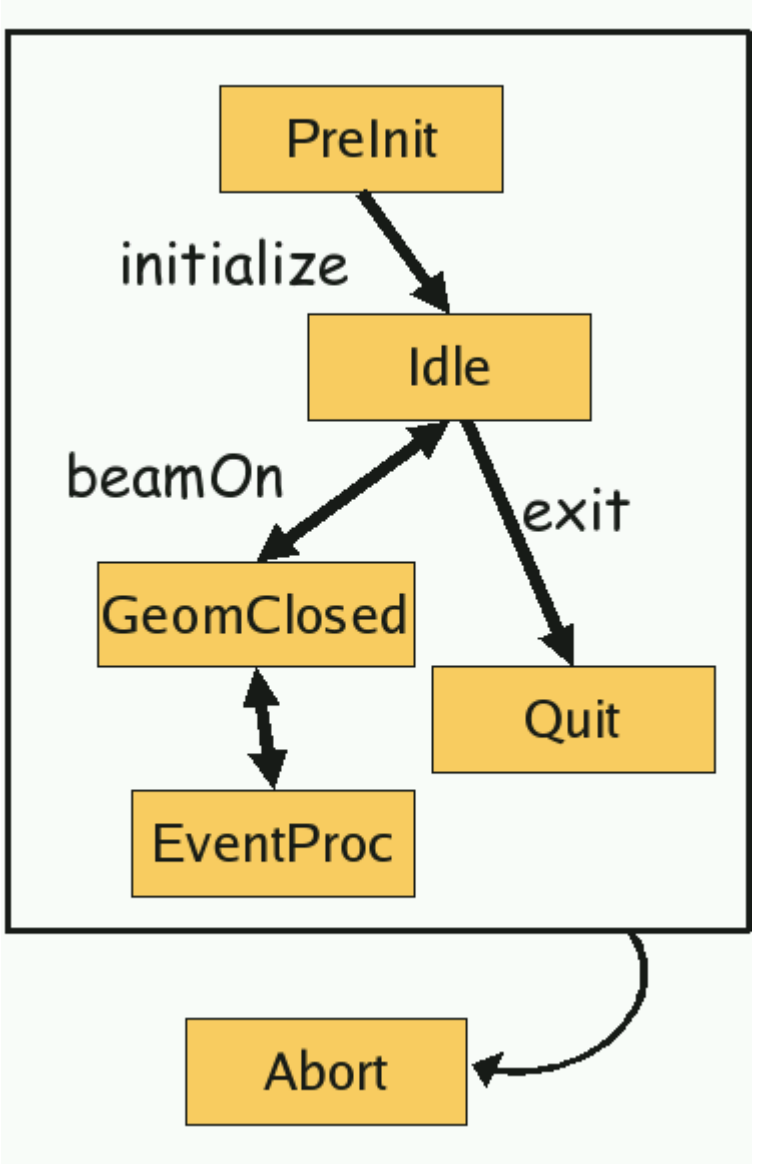
Step

- The step is the basic unit of simulation
  - Has two points (pre-step, post-step)
  - Contains the incremental particle information (energy loss, elapsed time, etc.)
  - Each point contains volume and material information
  - If step is limited by a boundary, the end point stands exactly on the boundary, but is logically part of next volume
  - Hence boundary processes such as refraction and transition radiation can be simulated



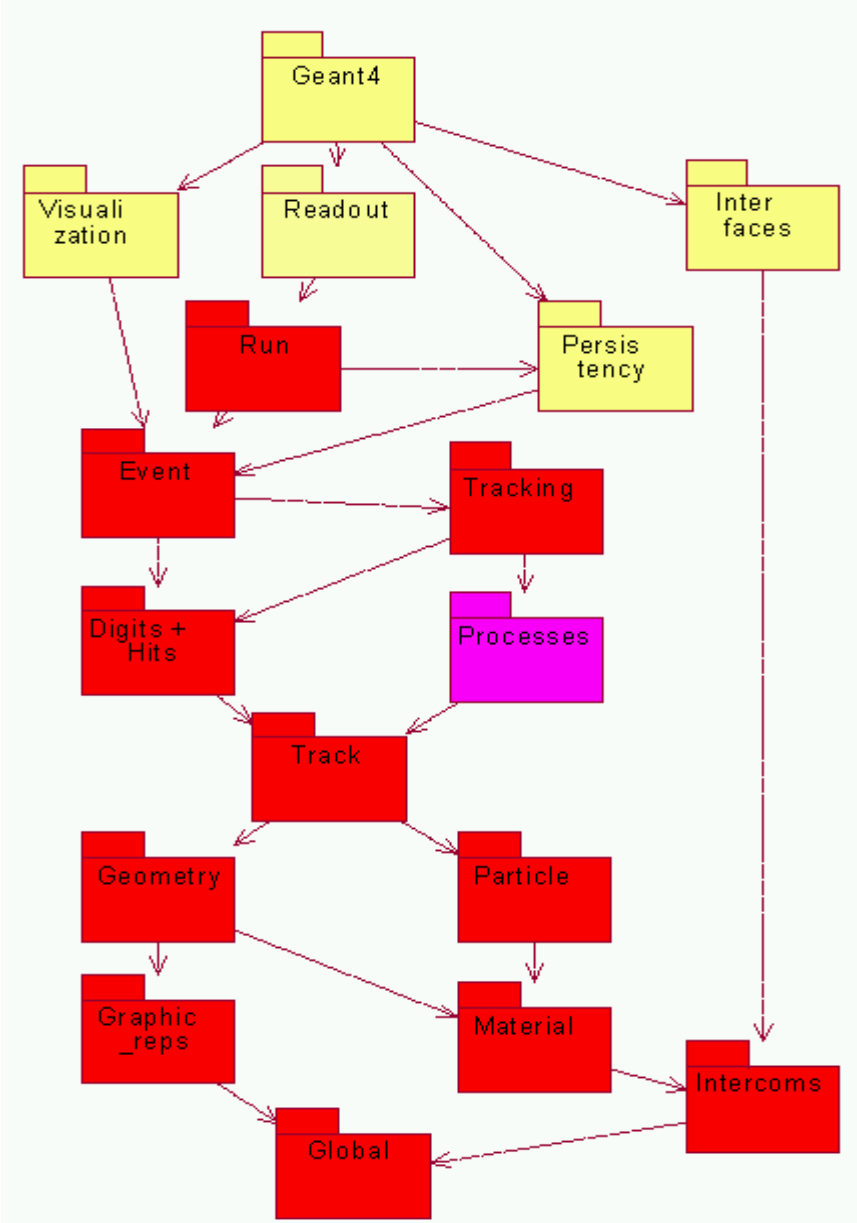
Geant4 as a State Machine

- Geant4 has six application states
  - G4State_PreInit: initialization, definition of geometry, material, particles and physics
  - G4State_Idle: may start run or modify geometry/physics for next run
  - G4State_GeomClosed: geometry is optimized, cross section tables updated, ready to process event
  - G4State_EventProc: an event is being processed
  - G4State_Quit: normal termination
  - G4State_Abort: fatal exception and program is aborting



Geant4 Categories

- Geant4 consists of 17 categories
  - Each is independently developed and maintained by a working group
  - Interfaces between categories are maintained by the global architecture working group
- Categories designed to minimize interdependence
  - Geant4 kernel consists of categories in red



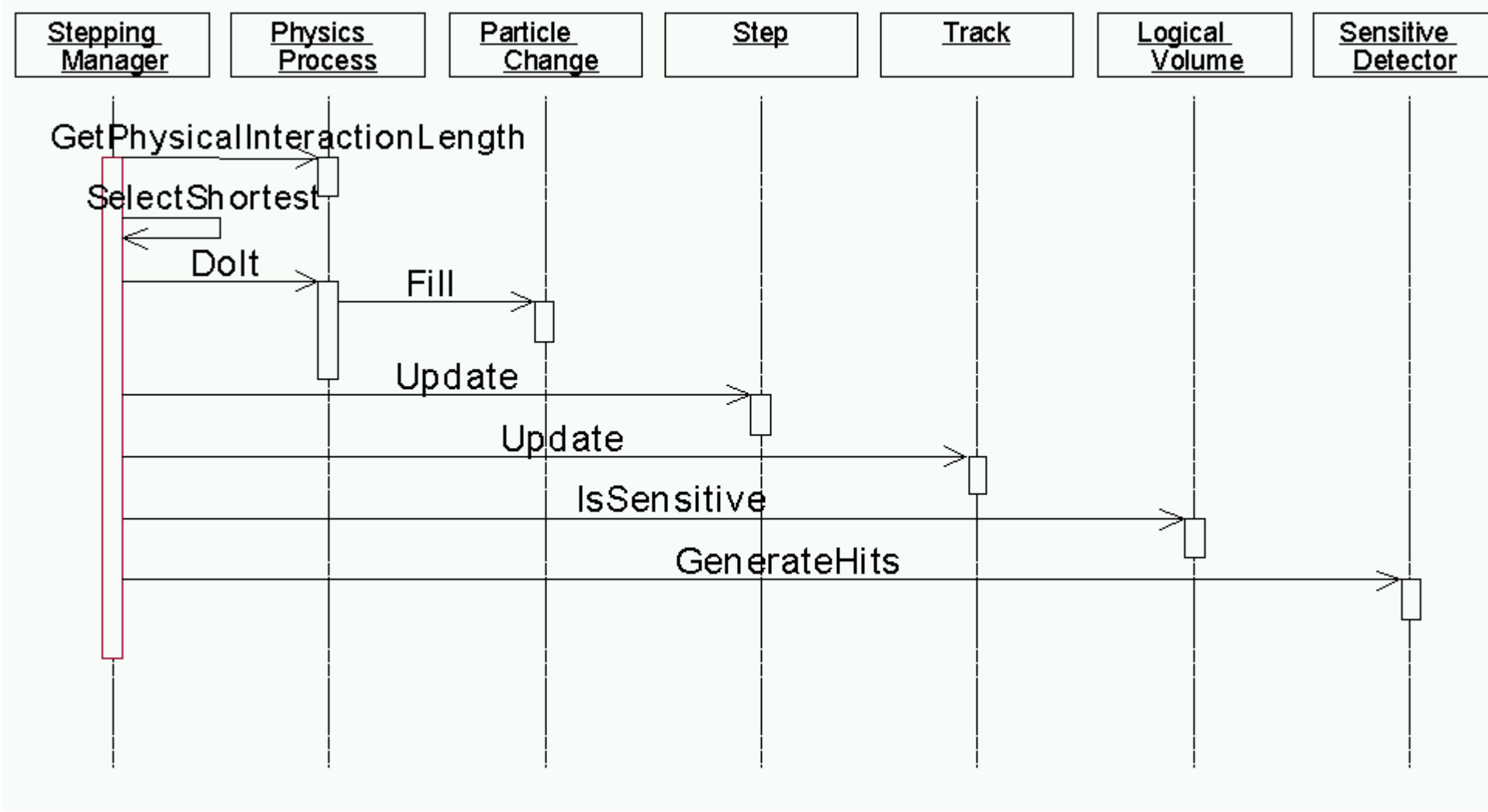
Tracking and Processes

- Geant4 tracking is general
  - It is independent of:
    - the particle type
    - the physics processes assigned to the particle
  - It enables all processes to:
    - contribute to the determination of the step length
    - contribute to any possible changes in physical quantities of the track
    - generate secondary particles
    - suggest changes in the state of the track (e.g. to suspend, postpone or kill)

Processes in Geant4

- All the work of particle decays and interactions is done by processes
  - Particle transportation is also a process; the particle can interact with geometrical boundaries and any kind of field
  - There is also a shower parameterization process which can take over from transportation
- Each particle has its own list of applicable processes. At the beginning of each step, all of these processes are queried for a proposed physical interaction length
- The process with the shortest proposed length (in space-time) is the one that occurs
  - The chosen process also limits the step size

How Geant4 Runs (one step)

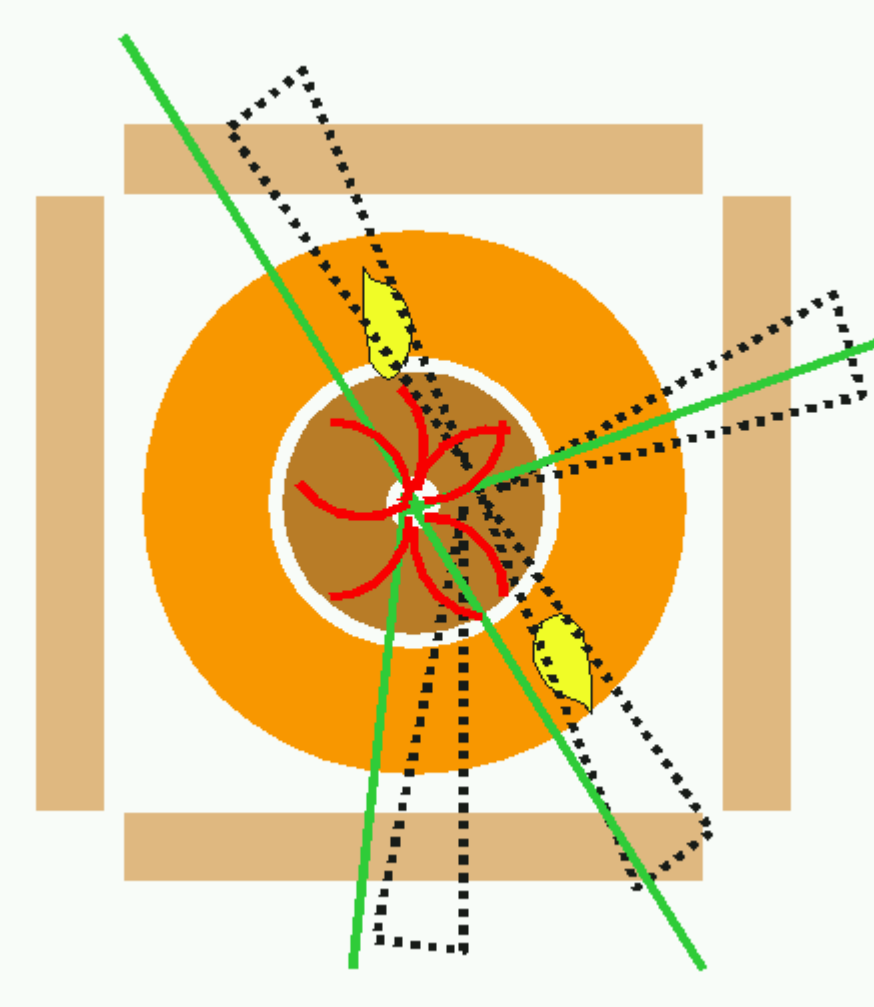


Cuts in Geant4

- A “cut” in Geant4 is really a production threshold
  - Only applies to physics processes which have infrared divergence
  - It is not a tracking cut
- An energy threshold must be determined at which discrete energy loss is replaced by continuous energy loss
  - old way:
    - track primary until cut-off is reached, calculate continuous energy loss and dump it at that point, stop tracking primary
    - above cut-off create secondaries, below cut-off add to continuous energy loss of primary
  - Geant4 way:
    - Specify range (which is converted to energy for each material) at which continuous energy loss begins, track primary down to zero range
    - above specified range create secondaries, below range add to continuous energy loss of primary

Track Stacking

- G4Track is a class object -> easy to suspend or postpone tracks
- Example:
  - Suspend tracks at entrance to calorimeter, i.e. simulate all tracks in tracking region before generating showers
  - Suspend “looper” tracks or abort or postpone them to next event
- Stacking allows prioritized tracking without a perfomance penalty
- Well thought-out prioritization or abortion of tracks/events make simulation more efficient



User Classes

- Use these classes to build your application on top of the Geant4 toolkit (class names in red are mandatory)
- Initialization classes
  - G4VUserDetectorConstruction
  - G4VUserPhysicsList
- Action classes
  - G4VUserPrimaryGeneratorAction
  - G4UserRunAction
  - G4UserEventAction
  - G4UserStackingAction
  - G4UserTrackingAction
  - G4UserSteppingAction
- Main() - not provided by Geant4

Describe Your Detector

- Derive your own concrete class from G4VUserDetectorConstruction
- In the virtual method Construct(),
  - assemble all necessary materials
  - build the volumes of your detector geometry
  - construct sensitive detector classes and assign them to the detector volumes
- Optionally you may define:
  - regions for any part of your detector (for production ranges)
  - visualization attributes of detector elements
  - magnetic (or other) fields

Select Physics Processes

- Geant4 does not have any default particles or processes
  - even particle transporation must be explicitly defined by user
- Derive your own concrete class from G4VUserPhysicsList abstract base class
  - define all necessary particles
  - define all necessary processes and assign them to the proper particles
  - define production threshold (cutoff) ranges and assign them to world volume and each region
- Geant4 provides many utility classes/methods to assist in the above tasks
- Example (educated guess) physics lists exist for EM and hadronic physics

Generate Primary Event

- For each event, user must define all details of initial particle
- Derive concrete class from G4VUserPrimaryGeneratorAction abstract base class
- Geant4 provides several ways to do this:
  - derive your own generator from G4VPrimaryGenerator
  - use provided generators:
    - G4ParticleGun (user provides number, energy, direction, type)
    - G4HEPEvtInterface, G4HepMCInterface (interfaces to high energy generator programs)
    - G4GeneralParticleSource (mostly for radioactivity)

User Action Classes

- G4UserRunAction
  - BeginOfRunAction (define histograms)
  - EndOfRunAction (fill histograms)
- G4UserEventAction
  - BeginOfEventAction (event selection)
  - EndOfEventAction (analyze event)
- G4UserTrackingAction
  - PreUserTrackingAction (create user defined trajectory)
  - PostUserTrackingAction
- G4UserSteppingAction
  - UserSteppingAction (kill, suspend, postpone track)
- G4UserStackingAction
  - PrepareNewEvent (reset priority control)
  - ClassifyNewTrack
    - invoked when new track is pushed
    - can set track as urgent, waiting, postpone or kill
  - NewStage
    - invoked when urgent stack is empty
    - event filtering

The main() Program

- Geant4 does not provide main()
  - However, many examples are provided in the Application Developers Guide
- In main(), you must:
  - Construct G4RunManager (or a class derived from it)
  - Provide to G4RunManager pointers to mandatory user classes:
    - G4VUserDetectorConstruction
    - G4VUserPhysicsList
    - G4VUserPrimaryGeneratorAction
- Other classes which can be defined in main()
  - VisManager
  - (G)UI session
  - Optional user classes

Setting Up a User Interface Session

- Geant4 provides several G4UISession concrete classes
  - Select the one that is appropriate for your computing envirnoment
  - In main(), construct one of them
  - Invoke its sessionStart() method
- UI sessions provided:
  - G4UIterminal – C- and TC-shell like character terminal
  - G4GAG – Tcl/Tk of Java PVM based GUI
  - G4JAG – interface to JAS (Java Analysis Studio)
  - G4UIBatch – batch job with macro file

Visualization

- Derive your own concrete class from G4VVisManager according to your computing environment
- Geant4 provides interfaces to several graphics drivers:
  - DAWN – Fukui renderer
  - WIRED – event display
  - RayTracer – ray tracing by Geant4 tracking
  - OpenGL
  - OpenInventor
  - VRML

Manager Classes

- Managers classes broker transactions between objects within a category and communicate with other managers
- They are singletons
- The user will have the most contact with G4RunManager
  - Must register detector geometry, physics list, particle generator to it
- Other manager classes:
  - G4EventManager – handles event processing, user actions
  - G4TrackingManager – handles tracks, trajectory storage, user actions
  - G4SteppingManager – handles steps, physics processes, hit scoring, user actions
  - G4VisManager – handles visualization drivers

System of Units

- Internal units system used in Geant4 is completely hidden from user code and from Geant4 source code implementation
- Each hard-coded number must be multiplied by its proper unit
  - `radius =  10.0 * cm;`
  - `kineticE = 1.0 * GeV;`
- To retrieve a number, it must be divided by the desired unit:
  - `G4cout << eDep / MeV;`
- Most commonly used units are provided , but user can add new ones
- With this system, importing/exporting physical quantities is straightforward and source code is more readable

Commands/Intercoms

- In Geant4, user can define commands and macros to run applications
- The Intercoms category handles the framework mechanism of defining and delivering commands
  - It is exportable to any other application
  - It is independent of other Geant4 categories
  - Uses strong type and range checking
    - C++ syntax : aCmd -> SetRange("x>0. && y>0.");
  - Dynamic command definition / activation
  - Commands can be hard coded or issued by (G)UI
- Macro files are enabled
  - Loop, foreach, alias, ...

The G4 Prefix

- For portability “G4” is prepended to raw C++ type names
  - G4int, G4double, ...
  - This way Geant4 implements correct type for a given architecture
- G4cout and G4cerr are ostream objects defined by Geant4
  - G4endl is also provided
- Some GUIs are buffer output streams so that they display print-outs on another window or provide storing/editing functionality
  - The user should not use std::cout, etc.
- Users should not use std::cin for input. Instead use the user-defined commands provided by the intercoms category
  - e.g. G4UIcmdWithADouble

Environment Variables

- To compile, link and run a Geant4-based simulation, the following environment variables must be set:
  - G4SYSTEM – operating system (e.g. Linux-g++)
  - G4INSTALL – base directory of Geant4 (where the compiled libraries are)
  - G4WORKDIR – where you run your application
  - CLHEP_BASE_DIR – location of the compiled CLHEP libraries
- Variables for physics processes (if those processes are used):
  - G4LEVELGAMMADATA – location of photon evaporation data
  - G4LEData -- location of cross sections for low energy EM module
  - G4RADIOACTIVEDATA – for radioactive decay processes
  - NeutronHPCrossSections – location of high precision neutron db
- Additional variables for GUI, Visualization, Analysis

Summary

- The Geant4 toolkit consists of 17 categories, each designed for minimal dependence on the others
- The largest unit of a simulation application is the run, which consists in turn of events, tracks, and steps
  - a track is a snapshot of a dynamic particle, not a trajectory
- Tracking and physics are carried out by processes
- Production thresholds and stacking allow for efficient simulation
  - Geant4 tracks particles down to zero energy
- User classes allow the simulation to be customized
  - user must build the apparatus, select the physics
  - commands allow user to communicate with simulation