G4Analysis: my view

Now a mature package with complete set of functionalities

I am using it everywhere in my applications and replacing native ROOT in some cases

- Because G4Analysis is MT/MPI friendly by default
- Because I do not need to care about having ROOT installed and configured (e.g. can be very challenging on Xeon Phi…)
- Because I prefer to use CSV format since I perform analysis with SciPy and Pandas

I've seen a very good feedback and strong interest in it in our tutorials

- Let's continue to extend its usage in our examples

General Comments 1

I've two applications that extensively use G4Analysis:

- ProcessLevel validation application (not part of G4 SVN yet)
- MonitoringExample (to be included in examples/extended/runAndEvent category for 10.3)

I could experiment with the power of G4Analysis:

- In both cases I select the output type (ROOT, CSV) at run-time instead of compile time.
  - Side Suggestion: can we add this functionality in the module itself?

General Comments 2

For ProcessLevel I created an application in which histograms or ntuples are defined and filled via UI command, thus allowing to add histograms without the need to write code

- Clearly this is application specific: what I did was to allow for binding a pre-defined "algorithm" to an histogram (or Ntuple column), so to say: "Fill Histogram #x with the value calculated by function XYZ()"

This triggered me about thinking of a way for Hits and Scorers to be stored in G4Analysis Ntuples in an automatic way, without the need to write user code

- E.g. use UI commands to say: "Write scorer XYX to Ntuple #x". Geant4 knows what a scorer is composed of and thus can create the Ntuple structure and fill it at run-time
- We actually did a test with Ivana and we managed to implement this in a relatively straight-forward way!
- Simplifies a lot data analysis for simple applications, no more core to write, just UI commands
  - Very useful to perform initial investigations and design a proper analysis

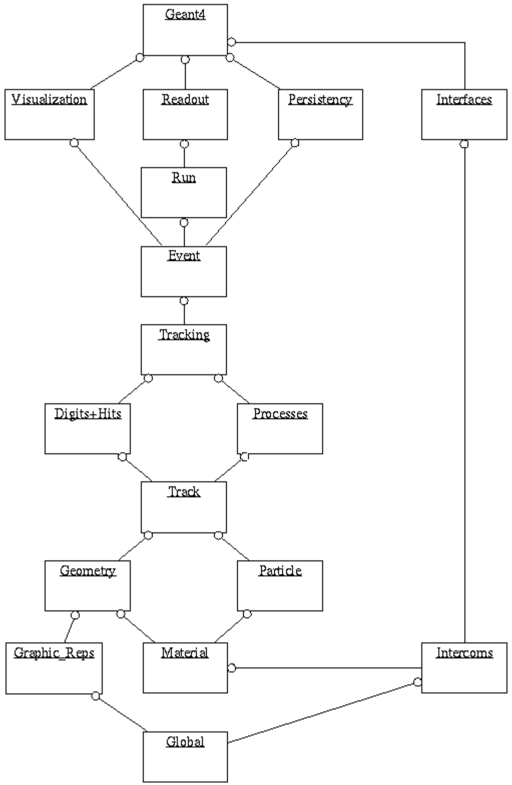
However...

We never pushed it SVN, why?

Because we could not clarify category dependencies and some general aspects like: who owns/controls the relevant objects? G4Run? G4RunManagter? User-code?

- We actually did not always agree on these aspects :-)

First thing to clarify



Look at our category diagram here:

- There is no mention at all of G4Analysis!

We do need a clarification:

- currently G4Analysis only depends on Global and Intercoms
- thus there is no problem if, e.g. Run category uses some functionality of Analysis (see Monitoring Example)

Proposal

Add to category diagram Analysis and clarify dependencies

With this in mind re-establish development for automatic storing of primitive scorers to analysis

- Clarify roles of G4Run, G4RunManager, …

If we are successful, I think we can even make a step further and allow **any** G4Hit to be stored in G4Analysis ntuples using the same mechanism used by Vis for picking

- Completely open to discussion,
- I propose however to think about this only after primitive scorers are done (and we can demonstrate we can do this)