Kernel Updates
From categorie: Run, Event, Digits_Hits, Particle, Intercoms

Andrea Dotti (adotti@slac.stanford.edu) ; SLAC/SD/EPP/Computing

Geant4 21st Collaboration Meeting – Ferrara, 12-16 September 2016

Improvements 1

Run Category:
- G4UnitsTable improvement in MT mode
  - Units now can be defined either in master thread, worker threads or in both. All cases are handled corrected
- New UI command: /run/workersProcessCmd to force threads to execute current command stack
- Change of behavior: an implicit "/run/workersProcessCmd" exists before threads quit to allow execution of UI commands after the last /run/beamOn

Event Category:
- Further improvements to GPS for MT support
- Particle gun can now shoot ions with floating level base

Digits_hits Category:
- Built-in scorers now use G4StatDouble by default for scorers accumulated for a run
  - Scores per event are still G4double

Improvements 2

Particle Category:

- Improve destructors of G4PrimaryVertex and G4PrimaryParticle (as requested by CDMS): solving issue and stack overflow for extremely large number of primaries
- G4IonTable, G4Ion now accept floating level base, e.g. Fe[176.0X]
- AME12 (Atomic Mass Evaluation 2012) is now the default DB for nucelus properties
  - But AME03 is optionally available
- Developments for muonic atom are underway
- Particles property updated to new PDG data

Intercoms enhancements

New ui commands in /control category:

- Compare two strings and perform action (execute file or single command): strif, strdoif
- Perform action depending on UI type: ifBatch, doifBatch, ifInteractive and doifInteractive

This set of new commands, extend the existing control flow commands: if, doif, add, subtract, multiply, divide, reminder, loop, foreach, …

- Quite rich set of possibilities for advanced cases: e.g. set G4Analysis histograms limits based on beam energy
- Calculate parameters from values

Example of macro file

```
/control/ifInteractive gui.mac


/control/getEnv EBEAM
/control/getEnv PARTICLETYPE


/control/multiply HISTOLIM .9 {EBEAM}
/control/strdoif {PARTICLETYPE} == "e-" /control/add HISTOLIM 1. {HISTOLIM}


/analyis/h1/create hSpectra "Spectra" 100 0. {HISTOLIM} {EBEAMUNIT}
```

Often we write bash scripts to modify a macro file before running a job, these can help to avoid this step

Non-planned features

A set of features presented in the next slides are not in the work-plan of kernel WGs, but have emerged from users during the year (ATLAS in particular)

- Since they are very useful we decided to implement them immediately

Multiple Sensitive Detectors

It is now possible to add multiple SD to the same logical volume.

This works creating a proxy that keeps a list of SDs and forwards calls to the list

To keep back-compatibility and to guarantee a zero CPU penalty for not-interested users, you need to explicitly activate this feature in user code:

```
#include "G4MultiSensitiveDetector.hh"
void myDetConstruction::ConstructSDandField() {
    //…
    auto msd = new G4MultiSensitiveDetector("aProxy");
    msd→AddSD( new mySD1 );
    msd→AddSD( new mySD2 );
}
```

Multiple user actions

Similarly to the case of SD it is possible to have multiple user actions. Exactly the same mechanism as before is set in place (explicit use via proxy class):

- `G4MultiRunAction`
- `G4MultiEventAction`
- `G4MultiSteppingAction`
- `G4MultiTrackingAction`

Not available for PrimaryGenerationAction and StackingAction because does not make sense in these cases: generation already supports multiple sources, multiple stacking actions can generate conflicts (e.g. one proposes to kill and one proposes to suspend)

More on multiple user actions

Special attention is needed in few cases:

- RayTracing
- ReverseMC

These features rely on "replacing" the user-defined RunAction with a special one and re-establishing the old one after a special run has been performed

- The existence of these use cases is the reason why we cannot make multiple user actions available by default, but explicit user code is needed