Decisions to take for G4 10.3

String models: FTF , QGS

Fork of String Models: FTF and QGS (1/2)

- Improvements of FTF model at thin-target level after G4 10.1 produce worse hadronic showers
  - True for G4 10.2 for energy response (too high)
  - True for G4 10.3 for lateral shapes (too narrow), mainly, with also slightly higher energy response, if we don't intervene...

- QGS fragmentation improved in G4 10.1; undergoing development in final-state
  - For G4 10.2, QGS was released exactly as it was in G4 10.1
  - For G4 10.3, QGS will be changed significantly...

Fork of String Models: FTF and QGS (2/2)

- Proposal : fork **parton_string/** into **parton_string_dev/**
  - **parton_string/** will stay stable, close to G4 10.1
    - For hadronic showers, used in nearly all physics lists
  - **parton_string_dev/** will keep changing (development)
    - For thin-target and development, used only in **FTFP_BERT_TRV**
  - Avoid collisions of symbols by using **stringdev::** namespace for symbols of parton_string_dev/
  - When FTF or QGS in the development area parton_string_dev/ is providing better thin-target and hadronic showers compared with the stable one in parton_string/ , the new one will be moved in parton_string/ and "frozen" (i.e. only bug-fixes are allowed), while further development will continue as usual in parton_string_dev/

Approved Decision

- Go ahead with the fork, using **g4partondev** as namespace
- The fork should be done quickly – by the end of next week if possible – to see whether we reach the limit on the number of symbols which are allowed in a single library by the Windows platform
  - If this is the case, then we need to split the library
- Investigate the use of the physics list factory to ease the Grid tests of the new string models
  - The ones in parton_string_dev/ which will be used only in the physics list FTFP_BERT_TRV
  - Alternatively, we could create new physics lists, e.g. FTFP_BERT_dev and QGSP_BERT_dev

ParticleHP
Environmental Variables

Environmental variables for ParticleHP (1/2)

- Currently, 5 environmental variables should be set:

```
export  G4PROTONHPDATA      =  /path/G4TENDL1.0/Proton
export  G4DEUTERONHPDATA =  /path/G4TENDL1.0/Deuteron
export  G4TRITONHPDATA      =  /path/G4TENDL1.0/Triton
export  G4HE3HPDATA             =  /path/G4TENDL1.0/He3
export  G4ALPHAHPDATA        =  /path/G4TENDL1.0/Alpha
```

- Proposal: introduce a new environmental variable

```
export G4TENDLDATA = /path/G4TENDL1.0
```

and there we find the subdirectories:

Proton/ , **Deuteron/** , **Triton/** , **He3/** , **Alpha/**

Environmental variables for ParticleHP (2/2)

- Get rid or keep the original 5 environmental variables?
  - Keeping them is convenient for users who want to use a different library *w.r.t.* the TENDL one distributed with Geant4, e.g.

    export G4PROTONHPDATA=/my/path-and-library
  - We could use the following algorithm:

```
if  ( G4TENDLDATA ) {
  if  ( ! G4PROTONHPDATA )       G4PROTONHPDATA=G4TENDLDATA/Proton
  if  ( ! G4DEUTERONHPDATA )    G4DEUTERONHPDATA=G4TENDLDATA/Deuteron
  if  ( ! G4TRITONHPDATA )        G4TRITONHPDATA=G4TENDLDATA/Triton
  if  ( ! G4HE3HPDATA )             G4HE3HPDATA=G4TENDLDATA/He3
  if  ( ! G4ALPHAHPDATA )          G4ALPHAHPDATA=G4TENDLDATA/Alpha
}
```

- For G4 10.4, proposal to get rid of all data libraries variables but a single one: **G4DATA_PATH**
  - Users can use their own libraries by pre-pending their paths to this environmental variables (similarly to LD_LIBRARY_PATH)

Approved Decision

- Keep the 5 existing environmental variables, and introduce the new one with the name **G4PARTICLEHPDATA**

- Investigate if it is feasible to introduce a switch in CMake to let the user to decide whether to download or not the ParticleHP data libraries
  - If yes, then do the same also for the NeutronHP data libraries
  - If not, then the ParticleHP data libraries must be downloaded separately by the users who need them