Update on LUXE GEANT4 Simulation.

Oleksandr Borysov
e-laser geometry

- Screen and Cherenkov detector moved backward and equipped with a support (John)
- Beam pipe til profiler and vacuum chamber in gamma spectrometer (Maryna)
- Lower position of tracker staves on electron side
• Another ECAL?
• CALICE (it is 20x20cm$^2$)?

How far below beam plane?
Vacuum chamber in gamma spectrometer
Vacuum chamber in gamma spectrometer

In case of difficulties in realizing complete construction a shorter version can be considered with substantial improvement in performance.
Performance of different IP - detector interfaces

- 50k electrons with energies 2.1, 2.5, 3.0, 4.0, … 16.0 (GeV).
- Vacuum;
- Beam pipe with Al windows;
- Beam pipe with Be windows;
- Vacuum chambers with Al windows of 2 mm;
- True MC comparison.

New G4 with 0.5 mm Al window for MC:
- w0_3000nm
- w0_5000nm
- w0_8000nm
4 GeV, vacuum chamber with different window thickness

29 µm in position uncertainty at 4 GeV corresponds to 0.5 MeV
Exporting Luxe geometry to gdml file

- Save the geometry in the beginning of the Run;
- it can be linked to simulation output;
- Can be imported back to Geant4 for check;
- Can be loaded to root for event display of simulated results;

```cpp
64  void RunAction::BeginOfRunAction(const G4Run*)
65  {
120   if (isMaster) {
121       if (fDumpGeometry) {
122           G4GDMLParser parser;
123           parser.Write("lxgeomdump.gdml", fDetector->GetphysiWorld());
124       }
125   }
```
Exporting Luxe geometry to gdml file

Seemed as a problem in G4GDMLWriteStructure.

Exporting assemblies to gdml file

The question is about exporting of assemblies to gdml file using G4GDMLParser::Write(...)
It seems that if a given volume contains imprints of several (e.g. two) different G4AssemblyVolume, then only one (the first) of them is exported to gdml <structure> section of that volume.
It looks like the choice of assemblies for gdml is made in recursively called G4GDMLWriteStructure::TraverseVolumeTree(...) 
First it selects assembly based on its ID (e.g. 1) and the content of class variable addedAssemblies:
L691 if(std::find(addedAssemblies.cbegin(), addedAssemblies.cend(), assemblyID) ==
addedAssemblies.cend())
in order to add it to <structure> section as assembly (e.g. <assembly name="Assembly_1">...)
and then based on the imprint ID (e.g. 1), which is implemented as independent on assemblyID:
L704 if(std::find(addedImprints.cbegin(), addedImprints.cend(), imprintID) == addedImprints.cend())
the imprint of the assembly is added to the volume description.

If there are e.g. two assemblies Assembly_1 and Assembly_2 which are imprinted in the same volume V1, then for the assembly Assembly_2 and its imprint 1 the second condition (L704) is false as imprint 1 was added for Assembly_1 and the imprint 1 of Assembly_2 is skipped.
At least this is how it seems to work for my geometry.

I tried a simple change:
std::vector<int> addedImprints substituted by 
std::map<int, std::vector<int>> addedImprints 
and then in L704 and L780 accordingly:
if(std::find(addedImprints[assemblyID].begin(), addedImprints[assemblyID].end(), imprintID) 
addedImprints[assemblyID].end()) ...) 
addedImprints[assemblyID].push_back(imprintID);
That seemed to solve the problem.
Problem 2343 - Multiple Assemblies aren't written to GDML fully

**Status:** RESOLVED FIXED

**Alias:** None

**Product:** Geant4

**Component:** persistency/gdml (show other problems)

**Version:** 10.7

**Hardware:** All All

**Importance:** P4 major

**Assignee:** Witold.Pokorski

**URL:**

**Depends on:**

**Blocks:**

**Reported:** 2021-03-03 20:50 CET by Laurie Nevay

**Modified:** 2021-03-17 19:04 CET (History)

**CC List:** 0 users

**See Also:**