Detector Geometry Description Tools in LHCb

G.Corti, D.Muller, B.Couturier
HSF Detector Simulation Working Group

Current status

- LHCb Geometry for Run 1+2 and for the upgrade described in the experiment specific *DetDesc* tool
  - We are now starting to validate the Upgrade detector description “as installed”
  - Checks and Visualization performed using LHCb specific tools
    (a.k.a OpenScientist based Panoramix that uses LCG 96b)
- Gauss application performs the conversion to Geant4 format
  - We can use the Geant4 validations tools after that transform

Port to DD4hep Compact

- Port to DD4hep (using the compact XML format)
  - Ongoing effort, some detectors are ported (VP, FT)
  - Checking how to integrate the DD4hep DetectorElements in LHCb Gaudi algorithms
  - Port and validation of the description in DD4hep challenging as the DetDesc description is being changed
- Run1/Run2 Geometry description will stay in DetDesc for the foreseeable future
  - DetDesc therefore staying in the stack

N.B. Port from DetDesc to DD4hep compact is a change of paradigm
(no easy automation of that step)

Issues so far

- ROOT TGeo was missing optical surfaces(and ways to store their properties)
    - Added to ROOT 6.18 partly and 6.20
    - Many thanks to the ROOT team for the developments
- Comparing DetDesc and TGeo is not trivial
    - DD4hep port went through a GDML export step that caused Assemblies to be removed (due to the LHCb Gauss application and to the way Assemblies are dealt with in Geant4/GDML) ⇒ Final CSG tree is not necessarily identical
    - Gauss export produces volume with identical names
      We use the G4GDMLParser option to make them unique by appending the Volume pointer
        - The export is NOT reproducible
    - Some recurrent problems:
        - different units, different origin positions for volumes, DetDesc objects without unique names

⇒ Geometry Validation and comparison tools are crucial to this effort

Opportunities

- Using TGeo opens the door to standard visualisation tools:
  - Possibility to export TGeo ⇒ three.js (via jsroot) ⇒ glTF (https://www.khronos.org/gltf/)
    - glTF can viewed/rendered in blender >= 2.80 (https://www.blender.org/)
    - Or reloaded in three.js to provide applications such as the HSF Phoenix (thanks to E. Moyse)
- ROOT7 Geometry tools will also allow rendering the geometry and events
  - Possible replacement for our current Panoramix tool
  - Gaudi process in the backend providing information to a web frontend
  - C.f the CHEP19 presentation on the web based widgets in ROOT (S.Linev) or the ROOT7 TEve (M. Tadel)
- Using TGeo opens the possibility to use VecGeom

Simulation

- Keeping support for the Run1+2 geometry description alive in Gauss (on Gaussino, CHEP2019 talk) for now
  - but might reevaluate this decision in the future
  - Could save/load GDML descriptions for all relevant conditions
- Mix and match geometry descriptions
  - DD4hep, DetDesc and GDML in Gauss
  - Support geometry description while in transition DetDesc ⇒ DD4hep
- For Upgrade geometry
  - We will use the converter from DDG4

Monitoring the effective geometry

We have tools in Gauss to collect radiation length and absorption length at given distances from the IP

- We can run them automatically and collect the results to compare effect of changes in either technology or description in our LHCbPR infrastructure

Conclusion

- Validating our Upgrade geometry using the LHCb DetDesc format
- Migrating to DD4hep Compact format
    - Change of paradigm
    - Using GDML export step is not easy due to the way we handle assembly and uniqueness of the names
    - All features in TGeo will be available as from ROOT 6.20
- Validation tools are crucial to this effort
    - We need to automate all validation steps to improve quality