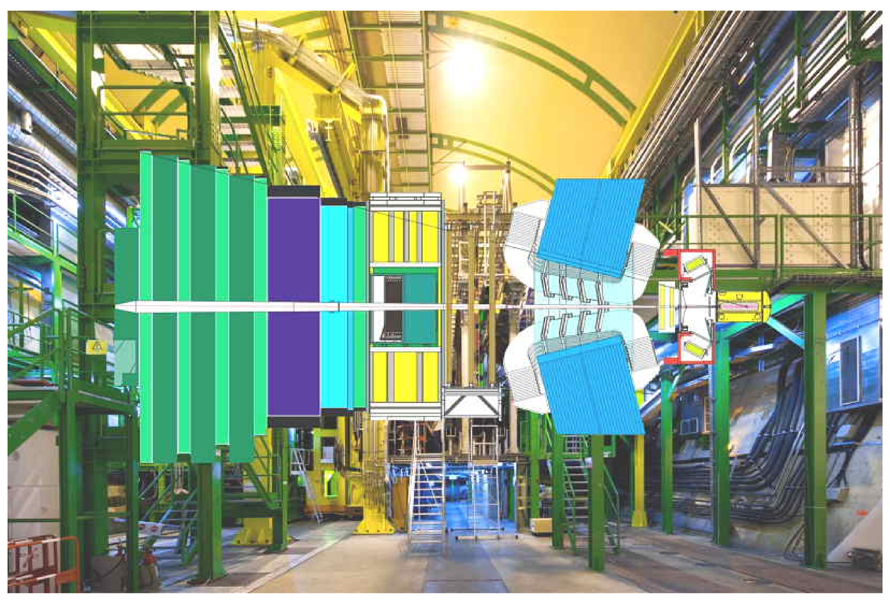
Perspectives for the migration of the LHCb geometry to the DD4hep toolkit

S.Borghi[1], C.Burr[1], M. Clemencic[2], G.Corti[2], B. Couturier[2], L.Grillo[1], M.Frank[2] on behalf of the LHCb collaboration
[1]The University of Manchester, [2]CERN

LHCb Geometry

Detector description framework in LHCb[1]

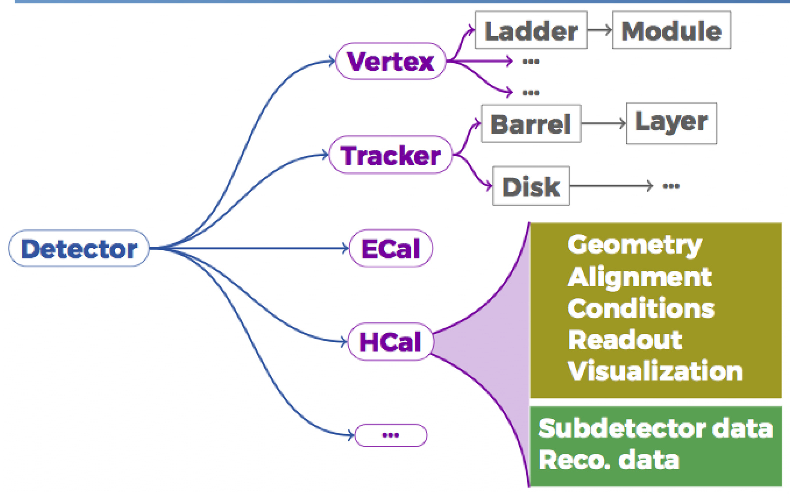


Stable framework for the last 15 years but…

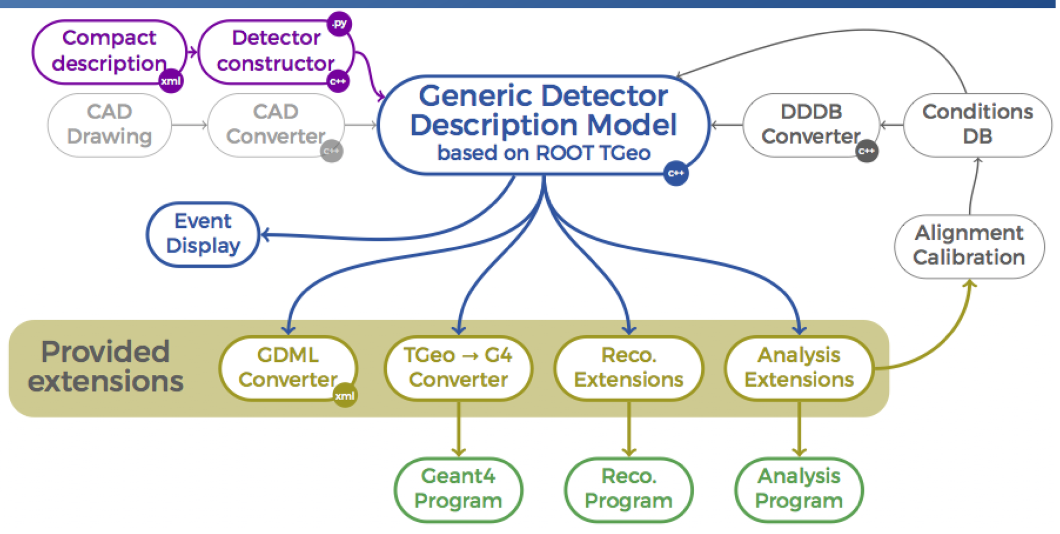
- Lack of effort to develop the framework
- Codebase suited to non multithreaded Gaudi
- Considerable room for improvements
  - Better integration of simplfied geometry
  - Redesign how we pass it to the Geant4 simulation
  - Integration with other simulation engines to be investigated

Custom geometry toolkit means that LHCb must develop all associated tools...

DD4hep Toolkit[2]



The generic description model

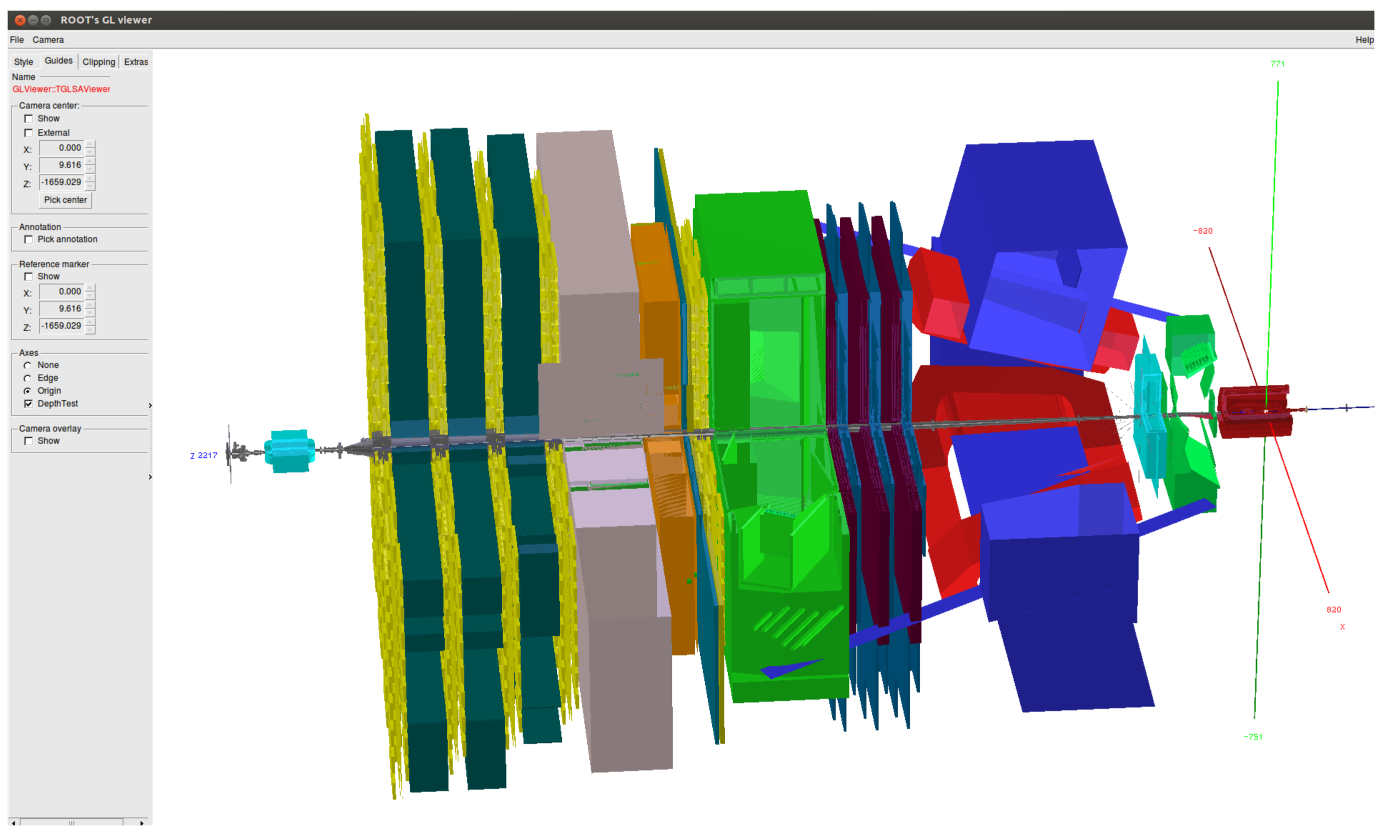


The toolkit components

DD4hep toolkit already used by other experiments (Linear Collider community, evaluation by CMS).
For more information see [4] and [5].

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement no. 654168.

Integration prototype



LHCb Geometry as loaded by the DD4hep DDDB Module (visualization using ROOT)

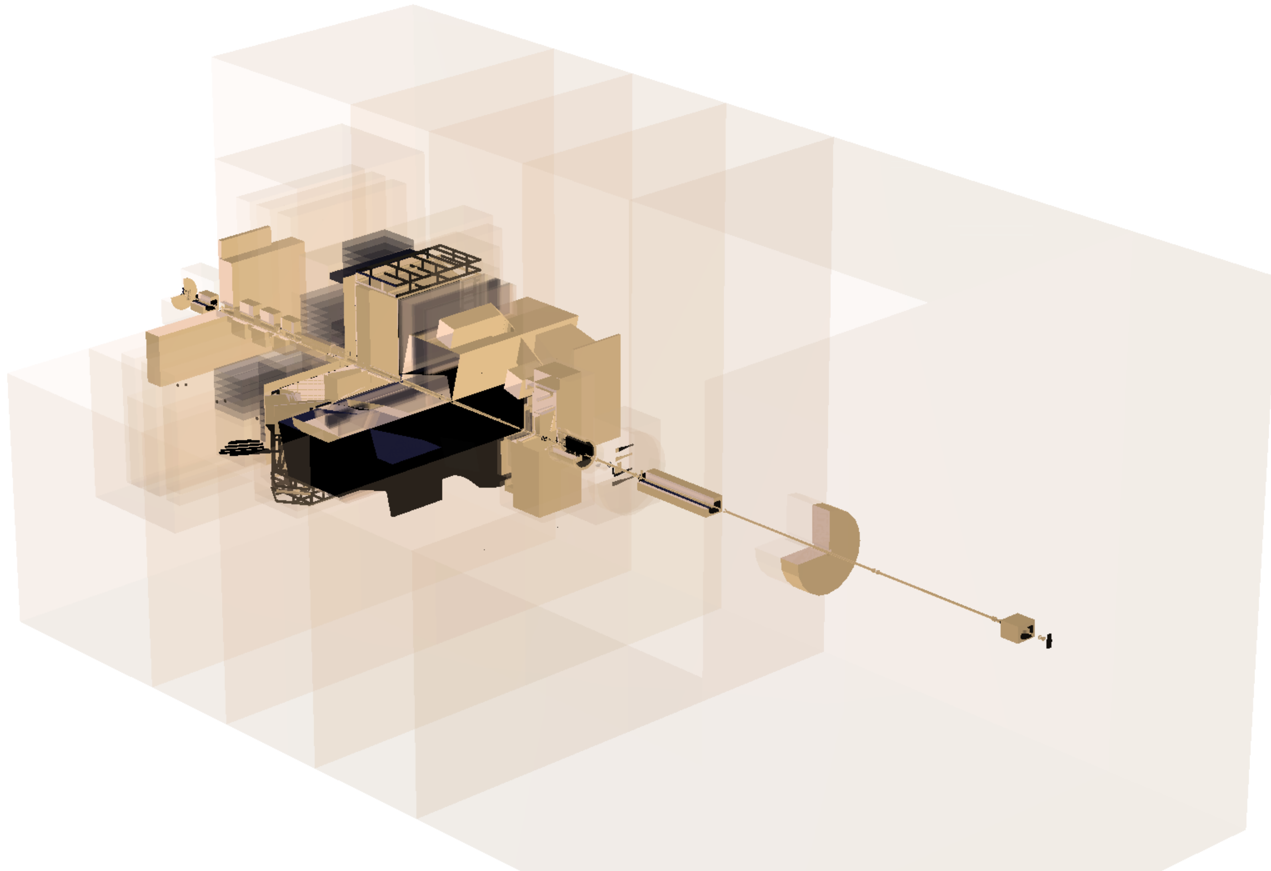
DDDB module from DD4hep

- Part of the DD4hep examples
- Allows loading the LHCb geometry … with some workarounds….

Current integration prototype allows to:

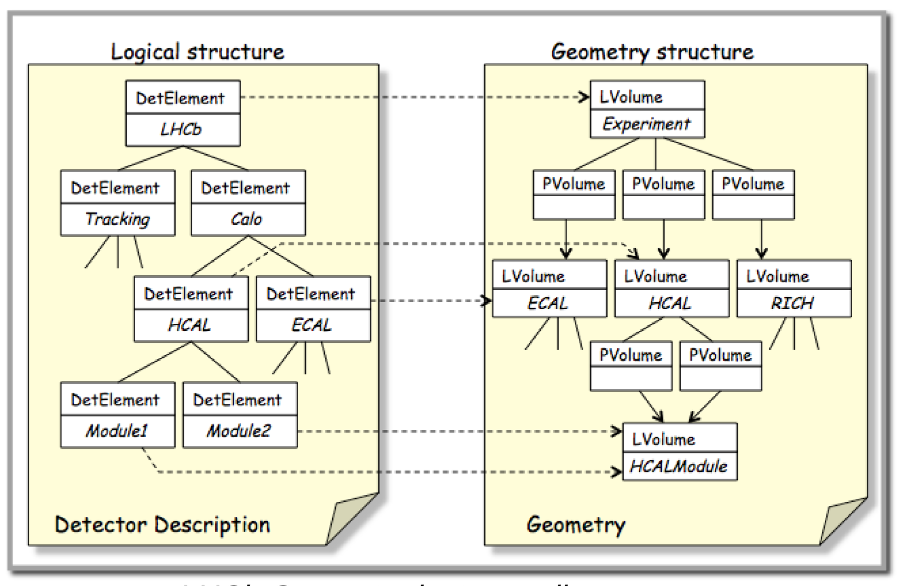
- **Compile** the DD4hep codebase in a way compatible with the LHCb software stack
- **Load** the LHCb Geometry with:
  - The LHCb code base
  - DD4hep

  within the same process
- **Compare** both representations in memory with custom scripts
- **Adapt the LHCb Geometry** with a local GIT repository
- **Push back** changes to the DD4hep code base

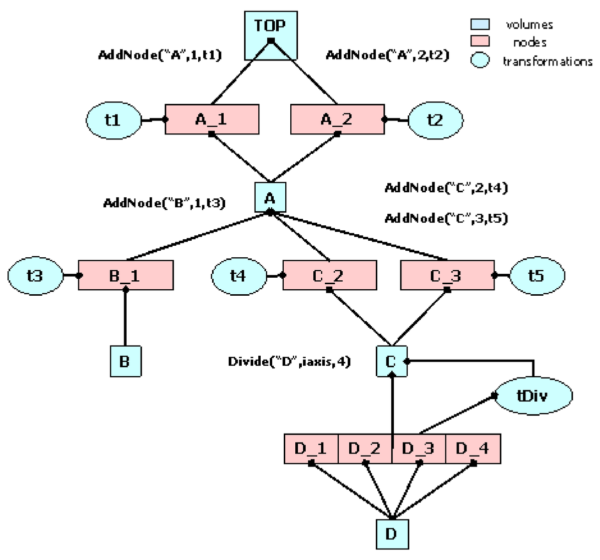


LHCb Upgrade Geometry as loaded by the DD4hep DDDB Module

Geometry validation



LHCb Geometry instance diagram



ROOT TGeo hierarchy in memory (from ROOT User's guide)

Geometry Class design

- Compatible class structure between the LHCb Geometry and the TGeo object model
- Volume libraries are also consistent
- Allows to compare *in memory* if both geometries are identical

Detector Elements classes

- C++ with custom classes for each sub-detector
- No automated port between the LHCb and the DD4hep representations

Geometry comparison done in two ways

- Hierarchical comparison of the volume trees
- Traversal of the detector on various paths to list volumes traversed and total radiation length

Conclusion of the current studies

- ***Good match between the two geometries***
  - Found/fixed minor problems with the DD4hep DDDB loader
- ***DD4hep DDDB loader is not a long term solution***
  - *Need to change LHCb's representation of the geometry*
- ***Need to validate (mis)alignment functionality***
- ***And integrate with the Simulation framework***

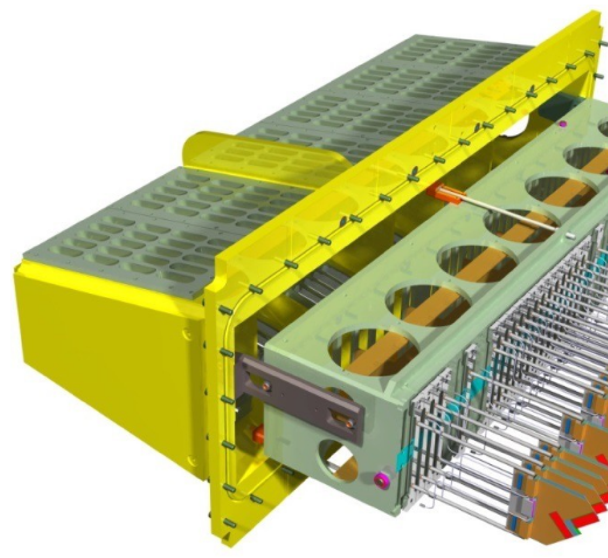
Alignment functionality

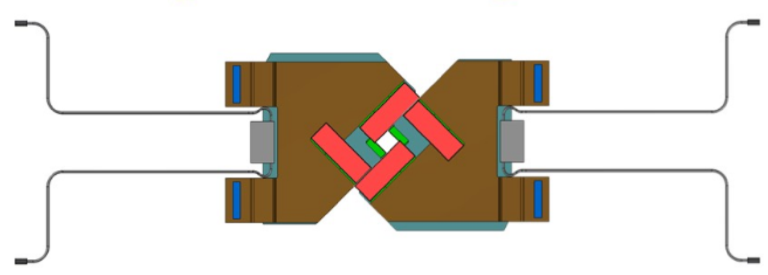
Detector alignment is a crucial functionality in HEP

- Currently working on a prototype of the LHCb Upgrade Vertex locator (Velo) Alignment functionality using the DD4hep prototype

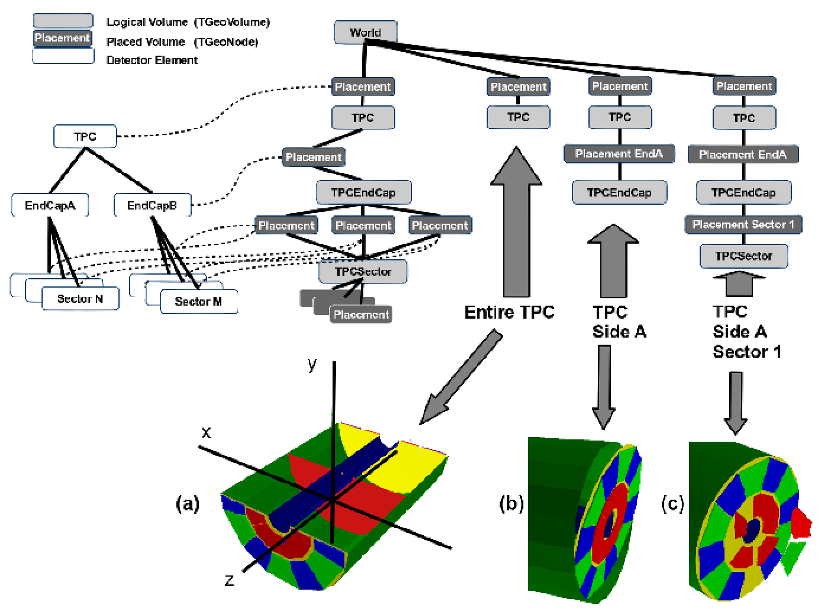
Prototype focusing on one detector: the LHCb VeloPIx

- Could check that the Velo sensors are placed correctly in the ideal geometry
- Investigating the functionality missing to implement the full Velo alignment





LHCb Upgrade Vertex locator. *Picture Copyright NIKHEF*



DDAlign functionality

Persistent format and future functionality

LHCb Geometry is not a good match with DD4hep

- ***Placements defined directly in the Geometry XML***
  - Works but is inflexble and difficult to debug
- ***DD4hep Compact XML approach***
  - *Volumes defined in XML*
  - *Placement done by C++ code*

Converting the LHCb Upgrade Geometry is a major task

- ***Fully automated conversion will be hard***
  - *Of course tools can help, especially for validation*
- ***Will require validation by all sub-detectors***
  - *Huge amount of work to follow up….*

Long term issues

Analysis of Run1 and 2 data does not stop at the upgrade

- ***We need to keep improving the simulation for the Run 1 and Run 2 dectector***
  - *Without keeping both the LHCb Geometry AND DD4hep code bases alive….*
  - *And without migrating the run 1 and 2 geometry to DD4hep*
  - *The LHCb Simulation application needs to be updated accordingly*[3]
- ***Can adapt the simulation framework to take GDML snaphots of the geometry***
  - *We need several snapshots depending of the data taking year*
  - *Simulation conditions will be loaded from the current database*

What do we gain ?

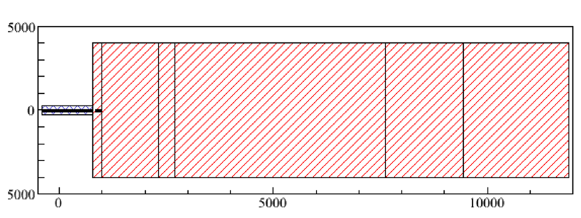
Using the full geometry hinders performance

- *Tracking in the full geometry too slow for the LHCb trigger*
- *Simulation represents ~2/3rds of LHCb CPU use on the grid*

LHCb current Geometry representation is very inflexible

- ***Either track and simulate with full detector***
- ***Or track in the (ultra) simplified geometry***

Current simplified geometry is a parallel representation done by hand with no links with the full geometry



LHCb Simplified Velo geometry

Need to work on a more flexible framework

- *Compact XML C++ constructors could give us that flexibility*
- *New design opens the door to such projects*

Custom geometry also means custom tools

- *LHCb developed the Panoramix event viewer usingOpenScientist*
- *Allows using related tools for visualization*

DD4hep is part of an ecosystem

- ***Uses ROOT TGeo as in-memory representation***
- ***Allows using related tools for visualization and checks***

- ***Porting LHCb to new geometry framework is a major endeavour, with major gains at hand...***
- ***Would get LHCb out of a dead end, and allow the experiment to share and collaborate on geometry representation and vizualization tools***
- ***Requires extremely thorough checks at all levels***

[1] **Detector description framework in LHCb**, S. Ponce, CERN. CHEP 2003, San Diego, USA, March 24-28, 2003
[2] **DD4hep Toolkit**, https://dd4hep.web.cern.ch/dd4hep/
[3] **Adopting new technologies in the LHCb Gauss simulation framework**, D.Muller CHEP 2018
[4] **New Developments in DD4hep**, M.Petric CHEP 2018
[5] **Conditions and Alignment extensions to the DD4hep Detector Description Toolkit**, M. Frank CHEP 2018