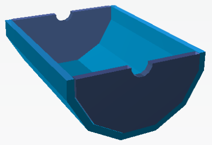
CLICdp/DD4hep

ECAL IN DD4HEP

AIDAN ROBSON AND DAN PROTOPOPESCU - GLASGOW, UK

CLICdp Optimisation Meeting | CERN, September 2014

Implementation/validation



The Glasgow Linear Collider Group has started porting and validating of the ECal model in DD4hep. We've contacted other relevant groups and collected pointers and howto.

Our task involves:

1) understanding the existing implementations/ frameworks
2) familiarising with DD4hep
3) implementing/updating the Barrel and EndCap ECal detectors in the new framework
4) validating the new implementation
5) providing the updated model to the detector optimisation group
6) goto 3 if changes are proposed

Along the way

We try to document every step of the process for other users/developers within the CLICdp and ILC communities, including new software implementations, proposed methodologies and new software tools.

We would also like to obtain feedback and check if what we're doing is right …

Checking Overlaps

We load the geometry in memory and display it using geoDisplay, e.g.

```
./geoDisplay compact/ECal.xml
```

To check overlaps or extrusions, from ROOT prompt we do

```
root [0]  gGeoManager->CheckOverlaps(0.01);
```

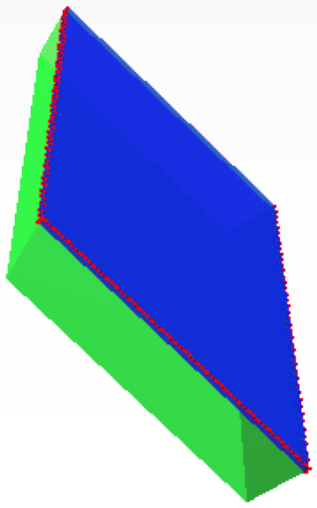
where the precision is given in `cm`. This will report any overlaps/extrusions:

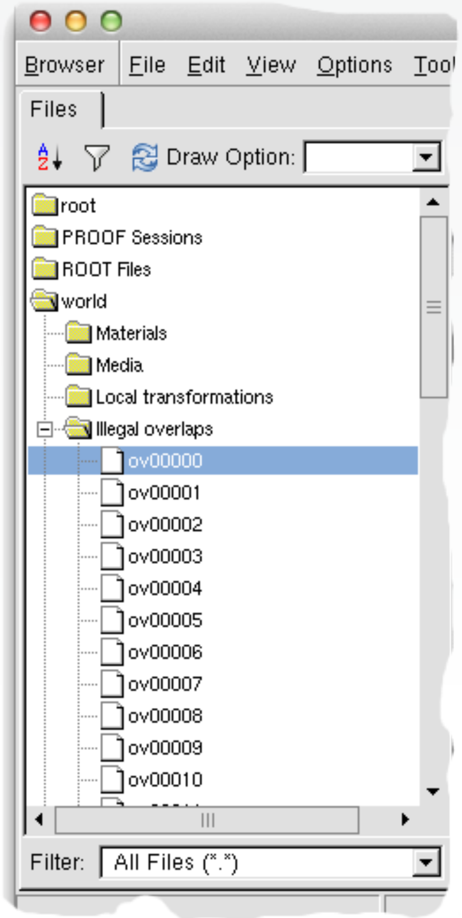
```
...
Info in <TGeoNodeMatrix::CheckOverlaps>: Number of
illegal overlaps/extrusions : 31
```

To visualise the overlaps/extrusions, navigate in the ROOT `TBrowser` to **world → Illegal overlaps** and you will be able to see a list of items.

The offending subvolumes are highlighted in red. Or one could use:

```
gGeoManager->PrintOverlaps();
```





Browsing and visualising overlaps in ROOT

Stress Tests

We found a tool for testing actual `TGeo` class implementations, which one could use to compare for example two descriptions produced via different frameworks, and spot if any differences were introduced.

Our wiki explains how one has to customise the class

```
$ROOTSYS/test/stressGeometry.cxx
```

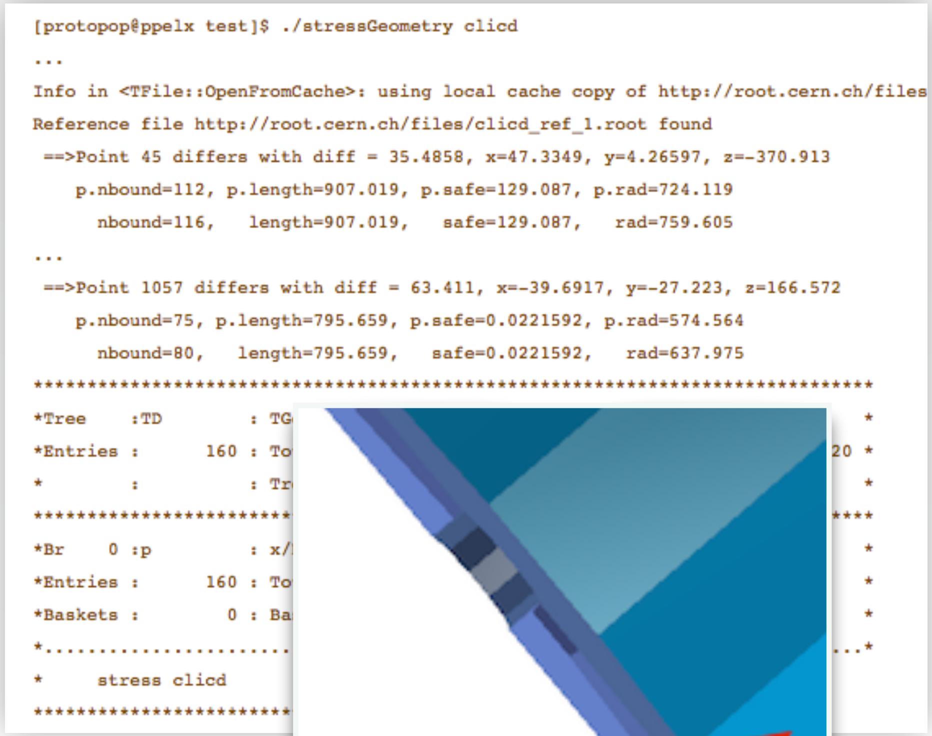
and add an new reference geometry, e.g `'myclicd'` which was created in DD4hep, and exported with

```
gGeoManager->Export("myclicd.root");
```

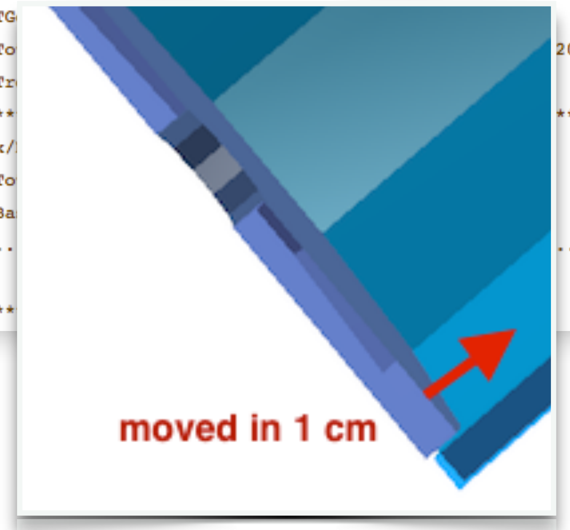
Then any further update to one's model can be verified against this reference with

```
./stressGeometry myclicd
```

The advantage of this tool is that it will test changes in geometry and materials including inert volumes, which would not be sensed by comparing hits via MC.



```
[protopop@ppelx test]$ ./stressGeometry clicd
...
Info in <TFile::OpenFromCache>: using local cache copy of http://root.cern.ch/files
Reference file http://root.cern.ch/files/clicd_ref_1.root found
 ==>Point 45 differs with diff = 35.4858, x=47.3349, y=4.26597, z=-370.913
    p.nbound=112, p.length=907.019, p.safe=129.087, p.rad=724.119
      nbound=116,   length=907.019,   safe=129.087,   rad=759.605
...
 ==>Point 1057 differs with diff = 63.411, x=-39.6917, y=-27.223, z=166.572
    p.nbound=75, p.length=795.659, p.safe=0.0221592, p.rad=574.564
      nbound=80,   length=795.659,   safe=0.0221592,   rad=637.975
******************************************************************************
*Tree    :TD         : TG                                                    *
*Entries :      160 : To                                                  20 *
*        :           : Tr                                                    *
************************                                                  ****
*Br    0 :p          : x/                                                    *
*Entries :      160 : To                                                     *
*Baskets :        0 : Ba                                                     *
*.......................                                                  ...*
*     stress clicd
************************
```



proposed

Geometry implementation workflow

Obtaining the latest geometry

The detector geometry is still being optimised, so it's a work in progress.

Best place to look is within the DDSim package, and the examples included in the DD4hep package.

Implementing or updating the latest ECal version in DD4hep

This involves using the latest software versions, and updating:
1) C++ driver(s) that actually build the module
2) XML description(s)
3) adding new material(s)

Validating the newly implemented ECal

This means (and/or):
1) comparing with a previous implementation of the <u>same</u> geometry
2) checking overlaps
3) comparing particle tracks
4) geometry stress tests

Committing the changes

Need a central repository for all CLICdp developers where:
1) updated geometries can be stored
2) subdetector overlaps can be checked
3) progress can be followed; ideas shared

Current status

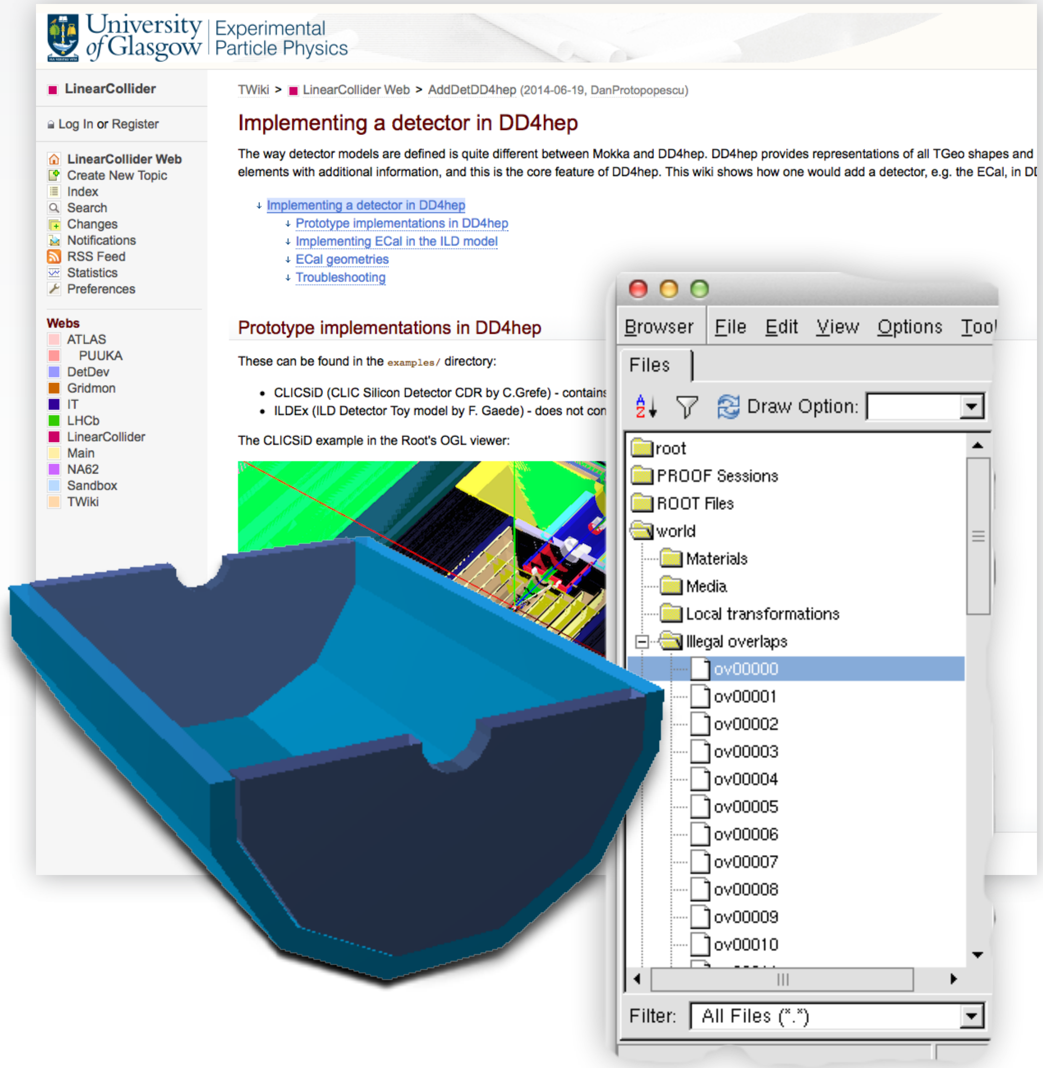
Implementation:

- adapted latest ECal drivers from S. Lu
- using latest s/w version(s)
- found and documented ROOT tools for geometry validation checks
- developed workflow (need feedback from other groups/developers)
- documented everything on our wiki

To do:

- find latest optimised version to validate against
- compare/share functionality with other subdetectors
- provide code for optimisation studies
- improve/update/rectify model once feedback is received

Need better coordination with other groups


University of Glasgow | Experimental Particle Physics

LinearCollider

Log In or Register

LinearCollider Web
Create New Topic
Index
Search
Changes
Notifications
RSS Feed
Statistics
Preferences

Webs
ATLAS
PUUKA
DetDev
Gridmon
IT
LHCb
LinearCollider
Main
NA62
Sandbox
TWiki

TWiki > LinearCollider Web > AddDetDD4hep (2014-06-19, DanProtopopescu)

Implementing a detector in DD4hep

The way detector models are defined is quite different between Mokka and DD4hep. DD4hep provides representations of all TGeo shapes and elements with additional information, and this is the core feature of DD4hep. This wiki shows how one would add a detector, e.g. the ECal, in DI

- Implementing a detector in DD4hep
  - Prototype implementations in DD4hep
  - Implementing ECal in the ILD model
  - ECal geometries
  - Troubleshooting

Prototype implementations in DD4hep

These can be found in the `examples/` directory:

- CLICSiD (CLIC Silicon Detector CDR by C.Grefe) - contains
- ILDEx (ILD Detector Toy model by F. Gaede) - does not con

The CLICSiD example in the Root's OGL viewer:

THANKS!

QUESTIONS ?
SUGGESTIONS ?

Visit http://www.ppe.gla.ac.uk/LC