dunetpc - Feature #14362

Add ROP support to 10 kt geometry

11/02/2016 01:53 PM - David Adams

Status:	Closed	**Start date:**	11/02/2016
Priority:	Normal	**Due date:**	
Assignee:	David Adams	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

We need to fill in the ROP methods for the 10 kt geometry as is being done for the 35t geometry in issue #14728.

Related issues:

Copied to LArSoft - Bug #14384: Copy of the Geometry object does not behave a...	**Closed**	**11/03/2016**

History

#1 - 11/02/2016 01:56 PM - David Adams

As with #14278, I would like to simplify and clarify the channel map class. I begin by copying ChannelMapAPAAlg to DuneApaChannelMapAlg so we can keep the old version around for comparison.

#2 - 11/03/2016 08:56 AM - David Adams

The 35t geometry test can now also be used for the FD geometry. Syntax is

```
$MRB_BUILDDIR/dunetpc/bin/test_Geometry_Dune35t dune10kt_workspace_geo/DuneApaChannelMapAlg
```

to test the new channel map and

```
$MRB_BUILDDIR/dunetpc/bin/test_Geometry_Dune35t dune10kt_workspace_geo/ChannelMapAPAAlg
```

to test the old channel map.

#3 - 11/03/2016 09:02 AM - David Adams

Gianluca:

While modifying DuneApaChannelMapAlg, I discovered that I get different results (when calling Geometry::WireCoordinate) if I make a copy of the wire geometry. To reproduce this, enable the #if statement at line 89. I don't know where the change is but if copying the geometry changes the original, we should probably delete the copy operator.

I am about to commit this change to dunetpc develop.

da

#4 - 11/03/2016 03:30 PM - Gianluca Petrillo

- Copied to Bug #14384: Copy of the Geometry object does not behave as the original one added

#5 - 11/07/2016 01:55 PM - David Adams

I have now filled in all the ROP methods for DuneApaChannelMapAlg and am about to commit to dunetpc develop.

Gianluca, could you run your test suite for 10kt using the DuneApaChannelMapAlg?

#6 - 11/07/2016 02:00 PM - David Adams

I would like to use the new class for both 35t and 10kt. I see the two have different sorters and my inclination now is to compile with both and select one based on an new fcl param. But I am open to other suggestions.

Are there any issues with the photon system? I will have a look at the code. Who should add to check that?

da

#7 - 11/07/2016 03:28 PM - Gianluca Petrillo

David Adams wrote:

> I would like to use the new class for both 35t and 10kt. I see the two have different sorters and my inclination now is to compile with both and select one based on an new fcl param. But I am open to other suggestions.

Is the sorter the only difference in the two cases? or does the channel mapping need to know which detector is being mapped for other reasons?
In the former case, I would rather suggest the caller to pass the right sorter to the mapping. Since the sorters are in the same class hierarchy, a DuneApaChannelMapAlg::SetupSorter(std::unique_ptr<geo::GeoSorterObject>&&) should do. In the end, sorting has little to do with channel mapping, and the more independent we keep them, the better.

#8 - 11/07/2016 05:27 PM - Gianluca Petrillo

I have run the test with no problem.
The tests, modified to use the new mapping for DUNE far and 35t detectors are in branch feature/gp_testsWithNew35tMapping of dunetpc (commit 8df7057), updated to the current develop branch (commit 659c885).

#9 - 11/08/2016 07:57 AM - David Adams

Gianluca:

I think I agree about keeping the mapping and sorting distinct. Simplest would be to make the geometry, mapping and sorting all simple services (i.e. no algorithm split) and then it would be straightforward to have the geometry find the fcl-configured mapping and sorting services by their interface. This desire to have service hierarchies is exactly why I don't like the algorithm model.

But you have gone down a different path and we need to decide which service or algorithm will instantiate the sorting class. I was going to do this in the mapping class but I think you suggest (and I agree) it would be better to put this in the helper service (DUNEGeometryHelper). The helper service is already ugly as it depends on all the different types of mappers and so adding (ugly) dependence on all the different sorters is best done there.

I suppose if we want to follow the split service-algorithm model (let me know if you have a better name), then we would create service interfaces for mapping and sorting and service wrappers for each concrete mapping and sorting algorithm. Then the helper would only need to depend on the interfaces. But this doubling in the number of classes is something I would very much like to avoid.

#10 - 11/08/2016 11:58 AM - David Adams

I have modified DuneApaChannelMapAlg to add a method to set the sorter and have modified DUNEGeometryHelper service to construct and pass the sorter based on the new fcl parameter GeoSorterClass. The test is modified to add this field to the arguments:

```
$MRB_BUILDDIR/dunetpc/bin/test_Geometry_Dune35t dune10kt_workspace_geo/DuneApaChannelMapAlg/GeoObjectSorterAPA
$MRB_BUILDDIR/dunetpc/bin/test_Geometry_Dune35t dune35t_geo/DuneApaChannelMapAlg/GeoObjectSorter35
```

The second invocation shows the APA mapper can now be used with the 35t geometry.

The fcl is not yet updated to use the new mapper class.

#11 - 11/08/2016 02:47 PM - Gianluca Petrillo

Note that the helper service does not need to depend on all mappings (and sortings). That is an implementation choice by DUNE. The alternative at this time is not fun either: one implementation for each channel mapping (plus sorting), and then each of them should check that the provided mapping is compatible with the loaded geometry[1], since there is the additional risk for FHiCL configuration to specify incompatible geometry and mapping.

[1] Here the design flaw is shining: the helper needs to query Geometry for the name of the detector geometry, but Geometry is not correctly initialised yet (at least, it's not sorted).

Diversion about the factorisation

The factorisation model I endorsed (for services I usually it service/provider model) has indeed as necessary consequence the doubling of classes, although it is non rare that one of them is just boilerplate (see e.g. lar::SimpleServiceProviderWrapper). But I don't see issues in implementing the structure you described with it, except that it requires more (trivial) code lines.
I haven't attempted the factorisation of ExptGeometryHelperInterface service, because it needs a careful design I did not have the time to undertake. The idea was similar to the one you write: a service for reading the GDML, a service interface for LArSoft geometry population (including sorting), another interface for channel mapping, and finally Geometry plus AuxDetGeometry as user interface. The transition to that structure would be a breath if experiments adopted a geometry configuration like: services: { @table::ExperimentGeometryBundle }, which I believe not commonly done. More likely, one would need to chase all FHiCL configurations and change them.

#12 - 11/09/2016 08:52 AM - David Adams

I have reorganized the tests in dunetpc/dune/Geometry/test so there are now separate tests for 35t and 10kt workspace.

#13 - 11/09/2016 10:12 AM - David Adams

I have added a test for protoDUNE and compared DuneApaChannelMapAlg with ChannelMapAPAAlg to verify the former can be used for protoDUNE.

#14 - 11/10/2016 01:38 PM - David Adams

I have modified DUNEGeometryHelper service so it now uses DuneApaChannelMapAlg by default for 35t, 10kt and protoDUNE. Note that the fcl parameter ChannelMapClass can be used to override this and use a different channel map, e.g. the now-obsolete ChannelMapAPAAlg or ChannelMap35OptAlg.

I verified that simulated raw data produced with the old and new channel maps (gen, g4, detsim, dataprep, analysis) give identical results for 35t and 10kt.

#15 - 11/10/2016 01:42 PM - David Adams

The above change is pushed to dunetpc develop.

Gianluca's geometry tests should now run without requiring any special configuration, i.e. without setting ChannelMapClass in DUNEGeometryHelper service. Please let me know either way.

#16 - 01/11/2017 12:14 PM - David Adams

- Status changed from Assigned to Closed

I assume this is all fine. Closing.