Geometry refactorization to support pixel readouts

Status report (Part 4 of N, where N is probably…hopefully…5)

Kyle J. Knoepfel

LArSoft coordination meeting

19 September 2023

Previous talks on geometry refactorization

- 9/20/22 **Status of Geometry service changes to accommodate pixel readouts**

  https://indico.fnal.gov/event/56265/

- 11/29/22 **Status of Geometry service changes to accommodate pixel readouts (part 2)**

  https://indico.fnal.gov/event/57355/

- 2/21/23 **Disentangling ChannelMapAlg from GeometryCore**

  or Status of Geometry service changes to accommodate pixel readouts (part 3)

  https://indico.fnal.gov/event/58509/

Motivation

- **LArSoft will support pixel geometries**
  - Adjustments to the Geometry service/system are required.
    - *We will separate readout-specific concepts from those of geometry.*
  - A few of us are meeting biweekly to determine how best to proceed.

Motivation

- **LArSoft will support pixel geometries**
  - Adjustments to the Geometry service/system are required.
    - *We will separate readout-specific concepts from those of geometry.*
  - A few of us are meeting biweekly to determine how best to proceed.

- **We have made significant adjustments to `larcorealg` to help us get there.**

- **But we're not done**
  - I will present the (revised) general tasks required to support pixel geometries.
  - Then we will talk about the status of the geometry system

Work plan to support pixel geometries

1. **Remove deprecated code** (done 11/10/22)

- Replace bare integer types with geometry IDs

```
728       -       const geo::CryostatGeo& cryostat = geom->Cryostat(cryo);
      728 +       const geo::CryostatGeo& cryostat = geom->Cryostat(geo::CryostatID(cryo));
```

- Keep only `geo::Point_t` and `geo::Vector_t` vector types

Work plan to support pixel geometries

1. **Remove deprecated code** (done 11/10/22)

- Replace bare integer types with geometry IDs

```
728       -       const geo::CryostatGeo& cryostat = geom->Cryostat(cryo);
      728 +       const geo::CryostatGeo& cryostat = geom->Cryostat(geo::CryostatID(cryo));
```

- Keep only `geo::Point_t` and `geo::Vector_t` vector types

2. **Adjust iteration patterns** (done 12/16/22)

```
342       -   for (const auto& tpcid : geom->IterateTPCIDs()) {
      342 +   for (const auto& tpcid : geom->Iterate<geo::TPCID>()) {
```

- Additional adjustments required to support PyROOT usage.

Work plan to support pixel geometries

3. Disentangle `ChannelMapAlg` and `GeometryCore` (done, *no PR yet*)

- Alters initialization sequence of `Geometry` and `ExptGeoHelperInterface` services
- Includes ownership adjustment of `GeoObjectSorter`

4. Extract `PlaneGeo` objects from `TPCGeo` (done, *no PR yet*)

Work plan to support pixel geometries

3. **Disentangle `ChannelMapAlg` and `GeometryCore`** (done, *no PR yet*)
    - Alters initialization sequence of `Geometry` and `ExptGeoHelperInterface` services
    - Includes ownership adjustment of `GeoObjectSorter`
4. **Extract `PlaneGeo` objects from `TPCGeo`** (done, *no PR yet*)
5. **Introduce readout geometry classes** (in progress)
    - Rename `ChannelMapAlg` → `WireReadoutGeom`
    - Rename `ExptGeoHelperInterface` → `WireReadout`
    - Refactor builders and sorters

Work plan to support pixel geometries

3. **Disentangle `ChannelMapAlg` and `GeometryCore`** (done, *no PR yet*)
   - Alters initialization sequence of `Geometry` and `ExptGeoHelperInterface` services
   - Includes ownership adjustment of `GeoObjectSorter`
4. **Extract `PlaneGeo` objects from `TPCGeo`** (done, *no PR yet*)
5. **Introduce readout geometry classes** (in progress)
   - Rename `ChannelMapAlg` → `WireReadoutGeom`
   - Rename `ExptGeoHelperInterface` → `WireReadout`
   - Refactor builders and sorters
6. **Validation** (not done)
   - Demonstrate that changes made above do not introduce physics changes.
7. **Support pixel geometries** (not done)
   - Skeleton interface developed by Tom Junk

`ChannelMapAlg` details (from last time)

- **`ChannelMapAlg` has been an abstract base class**

  Users supply their own implementations (or use the standard one from LArSoft)

  It maps readout concepts to physical geometry concepts

  It has contained a `GeoObjectSorter` (also abstract) whose implementations are used to sort the geometry objects upon geometry initialization.

`ChannelMapAlg` details (from last time)

- **`ChannelMapAlg` has been an abstract base class**

  Users supply their own implementations (or use the standard one from LArSoft)

  It maps readout concepts to physical geometry concepts

  It has contained a `GeoObjectSorter` (also abstract) whose implementations are used to sort the geometry objects upon geometry initialization.

- **`ChannelMapAlg` has not been directly accessible through any service**

  It is created through `ExptGeoHelperInterface` implementations

  It is owned by `GeometryCore`

`ChannelMapAlg` details (from last time)

- **`ChannelMapAlg` has been an abstract base class**

  Users supply their own implementations (or use the standard one from LArSoft)

  It maps readout concepts to physical geometry concepts

  It has contained a `GeoObjectSorter` (also abstract) whose implementations are used to sort the geometry objects upon geometry initialization.

- **`ChannelMapAlg` has not been directly accessible through any service**

  It is created through `ExptGeoHelperInterface` implementations

  It is owned by `GeometryCore`

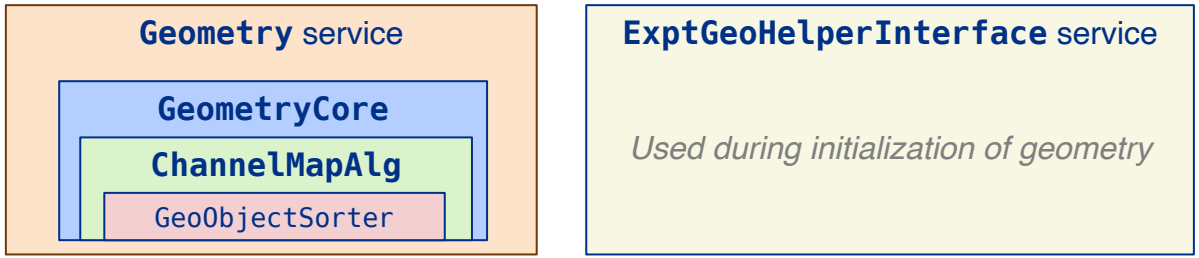
- **Many of `GeometryCore`'s functions directly forward arguments to `ChannelMapAlg` function calls.**

  Suggests a better factorization may be possible

`ChannelMapAlg` / `GeometryCore` refactorization

- It makes sense for a channel map to depend on the physical geometry
- It does not make sense for a physical geometry to depend on a channel map
- We are therefore removing `ChannelMapAlg` from `GeometryCore`

Now:

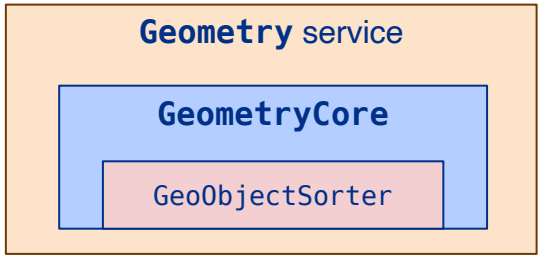


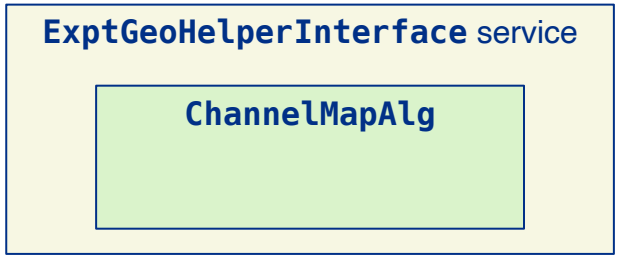
`ChannelMapAlg` / `GeometryCore` refactorization

- It makes sense for a channel map to depend on the physical geometry
- It does not make sense for a physical geometry to depend on a channel map
- We are therefore removing `ChannelMapAlg` from `GeometryCore`

Now:

Soon (done, *no PR yet*)**:**





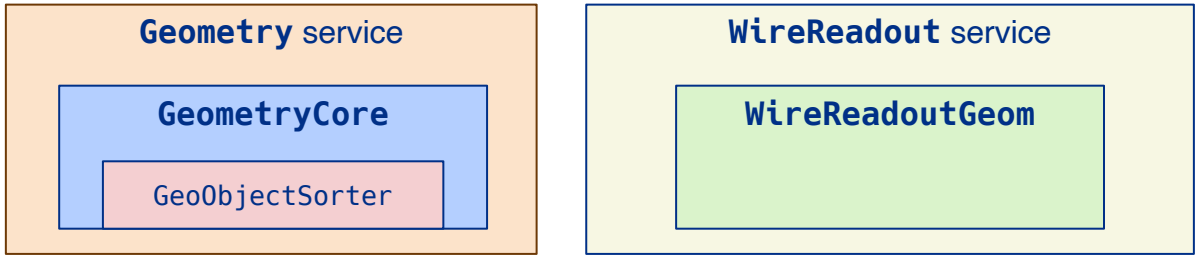
Fermilab

`ChannelMapAlg` / `GeometryCore` refactorization

- It makes sense for a channel map to depend on the physical geometry
- It does not make sense for a physical geometry to depend on a channel map
- We are therefore removing `ChannelMapAlg` from `GeometryCore`

Now:

Soon (done, *no PR yet*)**:**



Renamed

Fermilab

Factorization status

- **No PRs yet—this is just a status report of the implementation**

- Readout-based objects have moved from `GeometryCore` to `WireReadoutGeom`

  `TPCGeo` objects no longer own `PlaneGeo` objects

- Geometry builders and sorters

  Both physical and readout geometry systems will have their own builders and sorters (not yet implemented).

- FHiCL files will need to change, but it is not yet known to what extent.

  Will try to minimize breaking changes as much as possible

Other interface changes

- Changes in enumerations

```
- typedef enum coordinates {kXCoord, kYCoord, kZCoord} Coord_t;
+ enum class Coordinate {X, Y, Z};

- typedef enum driftdir {
-   kUnknownDrift, kPos, kNeg, kPosX = kPos, kNegX = kNeg
- } DriftDirection_t;
+ enum class DriftSign {Unknown, Positive, Negative};
```

- New struct for accessing drift-axis information.

```
struct DriftAxis {
  Coordinate coordinate;
  DriftSign sign;
};
```

- Simplifications to ID interface (e.g. `geo::PlaneID`).