

Prongs to be obsolesced

Eric Church, 23-May-2012

Prongs: a history

- Inheritance seemed sensible
- In particular, since Tracks and Showers shared some of the same properties: they are comprised of hits, they have a direction, energy,
- Idea was Track::Prong, Shower::Prong

Desires

- One's module could read up the output of the module that came before, and without knowing precisely what data it is working with, it could still manipulate that data and produce output.
- e.g., EventDisplay does this in places.
- T962/MINOS.h describes a class that may contain data or include extra MC members. Would like to be able to read either w.o. re-building whole T962.
- These work because there's no wish to output that data back onto the event or to pull more data via Associations.

Issues

- This is nicely general:

```
art::View < recob::Prong > prongListHandle;  
evt.getView(fProngModuleLabel,prongListHandle);
```

- These are not:

```
// Iterating over Prongs. But not Tracks!  
art::PtrVector<recob::Prong> prongIn;  
prongListHandle.fill(prongIn);  
art::PtrVector<recob::Prong>::const_iterator  
pprong = prongIn.begin();
```

```
// Using Assns to get the clusters for the  
spacepoints.
```

```
clusters = util::FindManyP<recob::Cluster>(prongIn,  
evt, fProngModuleLabel, cntp);
```

Not just that

- It's clear with all the Kalman and Bezier work that Tracks may need to carry state vectors and cov matrices and Bezier-ness. Showers do not. Methods to manipulate these things are not generalizable to Showers.
- etcetera.

Bottom Line

- It is troublesome to have persistent data classes -- meaning RecoBase objects that we write to the event -- be “polymorphic.” Which is to say that there may be good reasons to use such classes in analyze() modules. Not produce() modules.
- Chris Green confirms this is not supported in ART, for good reasons to do with CMSSW having wrestled with it for overly long and clunky solutions it required.
- LArSoft Statement: **As a matter of practice, a produce() module should know what data it's sucking up.** It's not enough to specify reading objects that enjoy some derivation of one to the other.
- Prongs will disappear in dev and going forward. Use Tracks or Showers.