LArSoft data product revision — phase II

Gianluca Petrillo

Fermi National Accelerator Laboratory

LArSoft Coordinators' Meeting, August 30th , 2016

The last data product review

The last data product revision effort was completed in January 2015

- 1D and 2D reconstruction classes were reconsidered
  - → a lot of small changes ensued
- `recob::Hit` and `recob::Cluster` were extended
- `recob::Track` was discussed, with no conclusion

Overview

LArSoft wants to start a *second phase* of revision:

1. adoption of a recommended data structure for physics vectors and for containers
2. definition of a data augmentation protocol
3. development of “view” tools to facilitate the use of data products
4. reorganisation of existing data products

Recommended data structures

The basic data structures stored in a data product deeply affect both interface and performance of LArSoft components:

- past discussion focused on
  - linear algebra classes
  - physics vector quantities
- a couple of meetings in June concluded with a plan of action
- I want to get to a physics vector class recommendation among:
  - `ROOT::Math::LorentzVector` and friends
  - `CLHEP::HepLorentzVector` & co.
- also, we should review the use of dynamically allocated collections:
  - reconsider `std::map`, `std::vector`, etc.
  - use `std::array` when/if serialisable (ROOT feature JIRA 8310)
  - if we want that for `std::tuple` too, we should open a request

Augmented data!

Two trends:

- larger data structures consume more memory
  - you end up loading lots of data even when you don't use it
- larger data structures consume more time
  - a producer is forced to fill all data even when it does not apply
- fragmented data structures are harder to use
  - need synchronisation between different data products

One solution:

- large data is fragmented is smaller storage units
- data is accessed via a unifying interface ("façade")

This will require some assumptions, that need to be agreed upon.
Overhead must be carefully assessed (*ideally, there should be none*).

Data as a view

LArSoft still lacks classes exposing different components of the reconstruction as a unity; for example:

- a track that contains its vertices, clusters and hits
- a particle set that includes showers and tracks
- an interaction, with vertices and particles (`PFParticle` tries)

Technically,

- might be implemented as an extension of the façade interface
- might benefit from Saba Sehrish's work on association navigation

But, more important than technical implementation:

Interfaces should talk the language of physics.

— Robert Kutschke

Reorganisation of data products

And, last and first:

- it's time to reconsider the choices from two years ago:
  - which data products did not age well?
  - what was left behind to be completed?
- the purpose, meaning and content of single data products should be discussed:
  - the track object is as bad as it was two years ago
  - `recob::Shower` needs a soul
  - `recob::Cluster` might be split
  - `recob::Vertex` needs uncertainty
  - and particle ID (MVA),
  - truth information (*nutools*)...

Summary

- starting now with the study of ROOT vs. CLHEP vectors
- the order of the rest is “in parallel” if possible
  - knowledge of what we want to do and what we can do are intermingled
  - but I’ll wait Fermilab Reconstruction group for some Kalman-fitter insight before reopening the track dance
- each and every item needs discussion and design
- will involve both Experiments and field experts
- this is a LArSoft project, but some requests could fall in *art* realm
- the project itself has a delivery time goal of January 2017

Backup