A Proposal to Inherit Configuration from Provenance

Alex Himmel

LArSoft Coordination Meeting
October 9th, 2018

https://cdcvs.fnal.gov/redmine/issues/20772

The Problem

- A problem I frequently run into in the photon simulation + reconstruction in DUNE is that services must be configured in consistent ways across different ART jobs.
  - Keeping consistent geometries at generation, reconstruction, and analysis.
  - Remembering pre-scale values used during G4 so they can be removed at the detsim stage.
  - Using the same photon library that was used during G4 to reconstruct energy.
- I have an existing work-around for the second example.
  - A script which can be run in advance to compare provenance to the fhicl being run.
  - But a user must know to run it for it to work!

A (Seemingly) Simple Solution

- But – all of the information to avoid these problems is available right in the ART file in the provenance already!
- My proposal is to extend the fhicl syntax to direct ART to pull configuration from a file's own provenance rather than the prolog.
- For example:

```
services: {
   LArPropertiesService: @file::g4.LArPropertiesService
}
services.LArPropertiesServices.ScintYield: 24500
```

  - would tell ART to configure LArPropertiesService the same way it was during the g4 stage of processing, but then override the ScintYield as usual.
  - Could imagine also defaulting to the last process if it is not specified.
- Obviously not 100% foolproof backwards compatibility
  - Breaks if parameters are added or removed, or their meaning changes, but we can plan around those things when developing services.

Not so Simple

- I discussed this plan with the Artists, which lead to a fun discussion and some insights.
- First, the plan I laid out has a fatal flaw:
  - As written, ART must be fully configured before the first file is opened since services can be used in the Source module.
- There is a potential short term solution: a wrapper script
  - We have a wrapper script which first fully expands the fhicl, looking for instances of calls to provenance.
  - If found, this wrapper runs config_dumper on the first input file and inserts the necessary segments into a the fhicl which is actually passed to the ART job.
  - In principle, we can name this wrapper "lar" and make it totally transparent to users.
- The ask – any volunteers to have a discussion of details of requirements for such a script with relevant experts from the LArSoft world.
  - I spend a lot of time worrying about getting correct results from people who are brand new to LArSoft.
  - Core developers and production experts likely have different needs and perspectives.

In the Long Run

- An important caveat: services which need this functionality likely shouldn't be services at all.
- The "right" model for how to handle information like this, as I understand it, is to have a producer insert it into the event record and have all processes access it from there.
  - The actual interface would be "overlay classes" which would act as a go-between for the data product and the developer.
- This, obviously, is a MAJOR undertaking
  - It involves serious rethinking of how many core LArSoft elements work.
  - But, I am told much of this work is necessary regardless to enable full use of multi-threading.