Wire Cell Toolkit Status

With Focus On LArSoft Integration

Brett Viren

(for WCT team)

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

FD SimReco

2017 Aug 17

Outline

Toolkit Overview & Recent Improvements

Wire Cell / LArSoft Integration Design

Configuration

Challenges & Issues

To Do & Summary

Overview of Wire Cell Toolkit

The toolkit consists of *software components* covering:

- Full **signal simulation** using **long-range and fine-grained detector response** and accurate **electronics response** functions.
- Full **noise simulation** using realistic, empirical and first-principles models.
- **Noise filter** to reduce (μBoone's) excess noise.
- **Signal processing** to deconvolve **detector response** and select **signal-ROI**.
  - Charge matching on all 3 planes, confirmed with μBooNE data.
  - Evaluated against full simulation to remove biases.

On the the way from Wire Cell **Prototype**:

- 3D ionization charge image reconstruct, pattern recognition, optical, ...

Recent Toolkit Improvements

Since last meeting, many important improvements to WCT:

- Prototype **signal processing** ported into toolkit (Xin, bv)
- Improved and validated signal processing to greatly **reduce charge bias** as a function of track angle (Hanyu)
  - → Now feel WCT sigproc gives solid foundation for 3D imaging.
- Refinement of sim's **signal induction** model (Hanyu)
- Corrections to **electronics response** model (Hanyu)
- Initial plan and steps for **simulation truth** recording (Brooke)
- Modeling and validating **2D field calculations** of detector pathology regions (Yichen, Mike).
- Start of a **validation suite** (bv)
- Improved **LArSoft integration** (bv) ... focus of this talk

Toolkit Overview & Recent Improvements

Wire Cell / LArSoft Integration Design

Configuration

Challenges & Issues

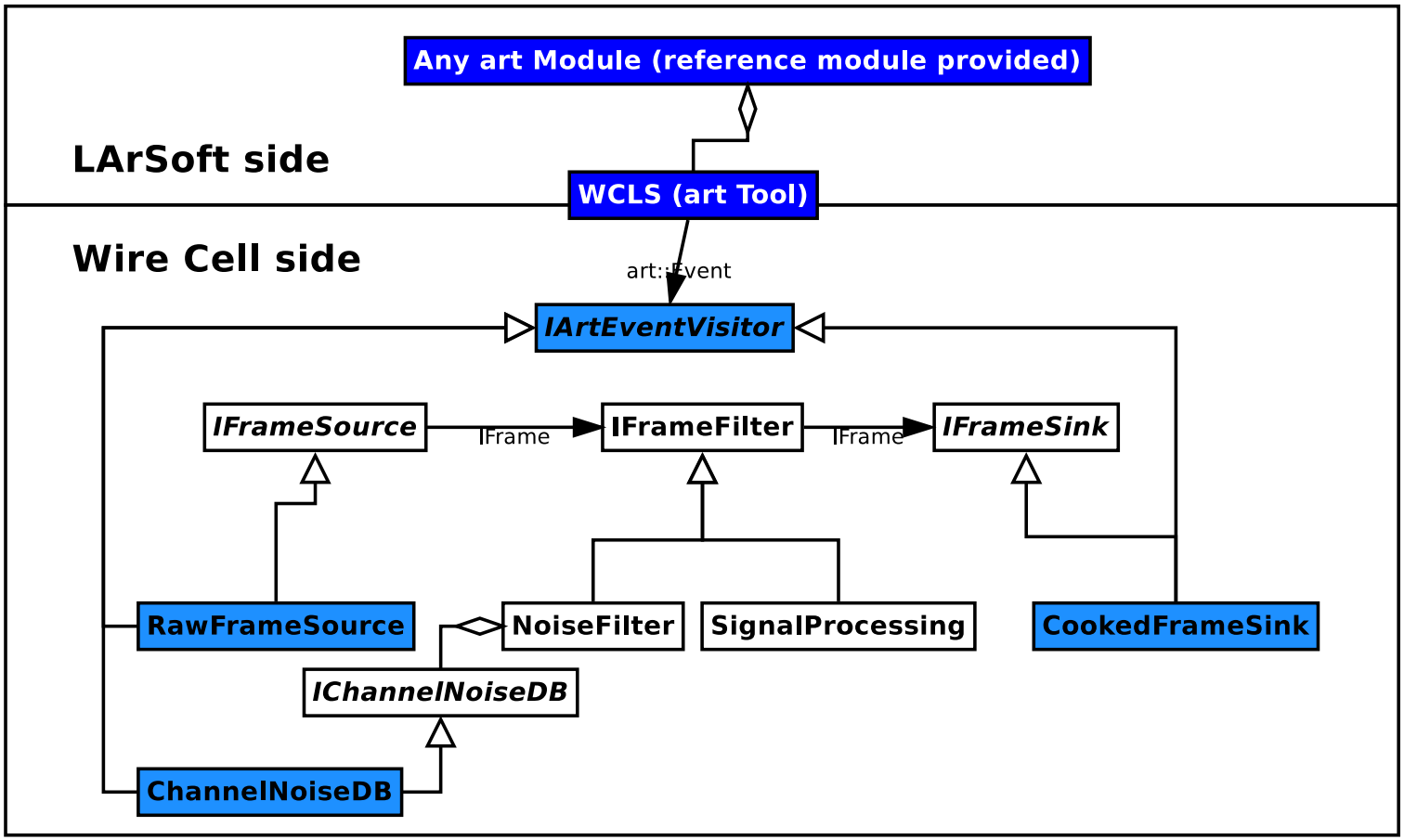
To Do & Summary

Initial Noise Filter Integration

- Dedicated MicroBooNE **noise filter** (Brian).
  - Got something working quickly, minimizing up-front effort.
  - It is kept as-is in `larwirecell/LarInterfaces`
- This helped to identify some issues:
  - Code implements a **tight coupling** between LS and WCT sides.
  - Requires detailed development updates as LS or WCT sides change.
  - Extending this pattern into the future will require a lot of coding, is error prone and tends toward copy-paste programming.
  - Lack of clear boundary leads to difficulty capturing full code, data and configuration state from LS-side for WCT-side debugging.

$\rightarrow$ The new integration design addresses these issues.

New WCT/LS Integration Design



WCLS integration code in `larwirecell` branch `feature/integration`.
Single *art* Tool for any/all entries to WCT. Reference *art* Module provided.
Dual-faced *converter components* to bridge LS/WCT data boundary.

Details: Integration Package `larwirecell`

- `Interfaces` directory holds any new **WCT Interface** classes that depend on *art*/LS data products
- `WireCellToolkit` is a reference *art* **Module** for running WCT. In principle, only one that is needed.
- `WCLS` *art* **Class Tool** providing main WCT entry point (equivalent to `wire-cell` CLI).
- WCT/LS **converter component** classes, eg:
  - `RawFrameSource` (`raw::RawDigit` → `WireCell::IFrame`)
  - `CookedFrameSink` (`WireCell::IFrame` → `recob::Wire`)
  - `ChannelNoiseDB` access to time-varying info for noise filtering.

→ Support for new WCT functionality only requires new *converter components* to be written, and only if new data products are introduced.

→ WCT code run in LS can be rerun, largely unchanged, in a pure-WCT environment which help experts to debug problems.

Configuration: *art*-side FHiCL

```
source :{
   module_type: RootInput
   inputCommands: ["drop *", "keep raw::RawDigits_*_*_*"]
}
physics :{ producers: { nfsp : {
  module_type : WireCellToolkit
  wcls_main: {
    tool_type: WCLS
    apps: ["Omnibus"]
    plugins: ["WireCellGen", "WireCellSigProc", "WireCellLarsoft"]
    configs: ["uboone-nf-sp.jsonnet"]
    inputers: ["wclsRawFrameSource"]
    outputers: ["wclsCookedFrameSink"]
    params: { detector: "uboone" }
   }
   p1 : [ nfsp ]
   trigger_paths : [ p1 ]
}
```

The `wcls_main` Tool accepts FHiCL parameter set which is essentially identical to the set of arguments accepted by the WCT CLI ("`wire-cell`").

Aside: *art* Configuration Support

art has some great support for defining configuration parameter set schema but **very few** LArSoft Tools/Modules/Services use it!

- Dev adds a little code to define each parameter in terms of **type, name and docstring**.
- Allows *art* to perform **input validation** on the FHiCL.
- Allows *art* to provide **command line help** messages.

```
$ art --print-description WCLS
...
[Description of all FHiCL parameters this tool takes]
```

It takes a little extra effort to add this support code, but I think it pays off for the user. Let's all use it!

Many thanks to **Kyle Knoepfel** for help on the LS mailing list to make use of these features!

Toolkit Overview & Recent Improvements

Wire Cell / LArSoft Integration Design

Configuration

Challenges & Issues

To Do & Summary

Integration Challenges

- Developers must build both WCT and `larwirecell` from Git.
- Building `larwirecell` requires WCT as `wirecell` UPS product.
- Dev's edit-build-test cycle requires injecting UPS product building!
- Done naively this would be complicated and painfully slow.

Solution: (still rather complicated!)

- Run FNAL's `buildFW` scripts used to produce UPS ecosystem (binary downloads but could potentially build some packages from source).
- Hack these scripts as needed, and esp. to define a new, temporary/bogus `wirecell` "release" version string.
- Clone, build and install WCT against UPS ecosystem and into `wirecell` UPS product area created above.
- Use `mrb` to build `larwirecell` against all the above.
- Hop back and forth between these last two shells, rebuilding/reinstalling

Works, but takes much effort to figure out and to replicate.
Many thanks to **Lynn Garren** for the help to make this work!

Memory Issue (not a leak!)

Observe memory footprint (VmSize) issue

1.5 GB	pure WCT job	no output file
2.0 GB	pure WCT job	simple TH2 output
3.0 GB	WCT + LS	no output file
4.5 GB	WCT + LS	*art* output file

- Memory footprint "should" be no more than 1 GB
  - naively: 2 $\times$ 300MB μB frames + $\sim$200MB config+lib data
- See evidence pointing to memory **heap fragmentation/inflation**:
  1. Allocate temporary working area
  2. Allocate output, free working area
  3. Job retains high water mark, with free'd "holes"

What do?

- More checking needed. Some heap layout viewer would help.
- Some mitigation ideas for WCT but seem much is beyond my control.
- For now, I leave it but I wanted people to be aware.

Toolkit Overview & Recent Improvements

Wire Cell / LArSoft Integration Design

Configuration

Challenges & Issues

To Do & Summary

To Do list

Near term:

- NF+SP: add writing of "bad regions" into `art::Event` (bv)
  - Probably as a `recob::Wire` collection.
  - Allow decoupling of NF & SP and expose intermediate data products to LArSoft
- Implement `SimEDep` $\rightarrow$ `IDepo` converter (Brooke, bv)
  - Relies on some LS development to make `SimEDep` available.
- Implement sim truth in WCT itself and saving to LS data product (Brooke, bv)

Longer term:

- Use WCT Sim+SP to investigate **protoDUNE data reduction**, implement corresponding LArSoft job config.
- Port **3D imaging** from WCP to WCT and develop its LS integration support (will need a converter for 3D image data).

WCT Release Roadmap

0.5.2 Support for MB NF module (past release).

0.6.x Support for NF+SP with new LS integration design.

- Expect to start on this after the collab meeting.
- Will also merge in `larwirecell`'s `feature/integration` branch for a LS release.

0.7.x Support for saving sim truth and multi-field response (eg, for "touching wires" in MB).

- Followed by `larwirecell` update with needed converters.

0.8.x 3D imaging in WCT.

Caveat: exact version strings may differ and priorities may change.

Summary

- Many improvements and refinements in WCT.
  - → **"Ready for 3D imaging!"**
- **WCT/LS integration is a solved problem**
  - All issues are at least understood.
  - NF+SP integration provides examples for future work.
  - New WCT functionality now easy to integrate and maintain.
  - New data converters only new integration coding needed.
- NF+SP release RSN. Then, sim followed by 3D imaging.
- Memory inflation issue may be a (broader?) concern.

fin

WCT Release Validation System

New `wire-cell-validate` package.

- Define small data set (MB, pD, WCT Sim).
- Run a number of *art* and `wire-cell` jobs.
- Also various plotting and comparison jobs.
- Generate JSON summaries and HTML reports.
- Package as Waf project for automation, parallel processing and idempotency.
- Rerun periodically and before releases to understand changes.