

# LArSoft - LArLite integration project organizational meeting

Erica Snider

*Fermilab*

April 9, 2015

# Agenda for the meeting

- Introduction(s)
- LArLite introduction
- Discussion
- Next steps

# Introduction

- LArSoft

- Primary simulation and reconstruction software
- Framework suitable for production processing

- LArLite

- Created as a “light-weight” development tool
- Has found additional use as a “light-weight” analysis tool
  - Aimed at stages after primary reconstruction completed

Kazu will explain in more detail the motivations and goals for LArLite

- For the purpose of further discussion, consider LArLite to be an example of a “light-weight” (LW) framework

# Introduction

(from discussion at the uBooNE computing review)

- LW analysis frameworks

(as discussed at the uBooNE computing review)

- Can be extremely useful after production processing
  - Many / most (?) experiments gravitate toward them for final stages of analysis
  - Some are extremely simple - e.g., root tuples
  - Some are more complex, capable, e.g.:
    - Use production (or production-like) data structures
    - Capable of (re-)applying some production(-like) tools and algorithms
- A poorly integrated LW framework can lead to significant additional effort
  - Data structures, tools, algorithms may be unnecessarily duplicated
  - Sometimes large overlap of functionality with little added benefit
  - Can lead to duplicated, uncontrollable ecosystems

These losses are unnecessary and should be avoided

That is the LArSoft goal of this integration project

# Integration

- We plan to “integrate” LArLite and LArSoft
    - Allow LArLite to be an integrated part of the LArSoft development environment
  - What does this mean?
    - Properly written algorithms and data structures in one can be used in the other with no modification
    - Data written by one framework can be used in the other
- CMS has succeeded in doing this for some fraction of their code

- What does that require?
  - Framework code and the application code are completely factored
    - Neither can depend on the other
  - An interface layer between the application and framework
    - For art, these are the art modules and art services

art developer guidelines suggest exactly this factorization

# What are the issues?

- Factorization of art / LArLite from algorithm code
  - For LArSoft
    - Define algorithm classes for simulation and reconstruction algorithms
    - Pass in services and data products, get data products out
    - Define art services that contain the application services or derive from them
    - Deal with art::Ptr and art::Assn

Mostly straight-forward, but a lot of work

- About 240 modules to examine...
    - This factorization is a goal (of several) of the on-going architecture project
  - For LArLite
    - Remove TObject inheritance from persistable data structures
      - Currently modify LArSoft data products to include this inheritance
    - Other things to be determined...

# What are the issues?

- External dependencies
  - art uses a number of external products
    - fhicl
    - cetexceptions
    - etc
  - Many of these used inside modules, algorithm and service code
  - Need a strategy to deal with these
    - Consider case by case
    - Adopt or virtualize via a common LArSoft - LArLite interface?

# What are the issues?

- Policy considerations
  - The result must meet the existing requirements of each system
    - Want MicroBooNE to be successful with whatever framework(s) they choose
    - Need clear statements as to what the LArLite requirements are, what the intended use cases are
  - Goal is to enable / provide complementary capabilities
    - Does not mean orthogonal
    - SCD is not committing to support of a second production processing system
  - No change in basic LArSoft support policies after the integration
    - MicroBooNE will be solely responsible for LArLite
    - LArSoft project will retain current responsibilities
    - Will need to cooperate around changes that might affect integration
  - A possible outcome: SCD supports an API to LW frameworks
  - Other policy questions TBD?

Get an understanding, agreement on policy issues early.



# Summary

- Available effort
  - SCD (total of 0.5 FTE)
    - Chris Jones
    - Marc Paterno
    - et al. TBD
  - From MicroBooNE (0.? FTE?)
    - Kazu Terao
    - Bill Seligman
    - Other members of MicroBooNE offline, LArLite team TBD
- Expect project to last approximately six months(?)