# LArSoft

Erica Snider

*Fermilab*

*LArSoft core support team:*

Vito di Benedetto

Lynn Garren

Patrick Gartung,

Mark Mengel, Gianluca Petrillo,

Vladimir Podstavkov, Erica Snider

LArSoft Assessment and Requirements Workshop
Oct. 19—20, 2015
Fermilab

## LArSoft project / collaboration

- A collaboration of experiments, Fermilab, other stakeholders
- Goals
  - To provide an integrated, art-based, experiment-agnostic set of software tools to be used by multiple LAr neutrino experiments to perform simulation, data reconstruction, analysis
  - By developing common data structures, algorithms, architecture
    - Reduce the cost of developing, supporting and maintaining the simulation and reconstruction software for collaboration members
    - Avail ourselves of the widest possible community of experts and developers to reconstruction, simulation solutions
    - To propagate knowledge directly detector-agnostic via code

# Legacy LArSoft requirements

- Understand why we are were we are now
- Some history
  - Shared core software originally developed for ArgoNeuT and MicroBooNE
    - Effort initiated, led by Brian Rebel and Eric Church
    - Recognized common features for LArTPC technology
    - Allows leveraging effort across multiple experiments to solve common problems
  - A set of agreements required to achieve this sharing
    - Definition of core software to share in common
    - Common set of data abstractions, interfaces
    - Common Interfaces to configuration data
      - Detector properties, LAr properties, etc.
    - Common workflow elements to connect data abstractions
      - The steps in the reconstruction that create higher-level data products from lower-level
    - External dependencies
    - Common / compatible build systems, version control system

## Legacy LArSoft requirements

- All major architecture, workflow elements grew from this early collaboration
  - Raw data calibration and deconvolution
  - Hit finding
  - Cluster-finding
  - Track-finding
  - Shower-finding
  - Vertex finding
  - Momentum estimation and particle ID
- Not included in these early requirements
  - Specific physics requirements
    - Left entirely to the experiments to define and manage
  - Comprehensive set of unifying design principles and guidelines

## The collaboration now

- Some of the core principles of this collaborative effort
  - Development and priorities driven entirely by the experiments
    - Core project team supports that development
  - Collectively agree on the unifying principles
  - A distributed community working bottom up – all are welcome
  - Encourage innovation and new ideas, provide interfaces to alternative algorithms and approaches, integrate code for use by the collaboration
    - Pandora
    - “Wire-cell” approach
      - Recently introduced by Xin Qian, Chao Zhang, Brett Viren
      - Working to integrate fully at

## Core LArSoft project

- Focused on ensuring the success of the sharing paradigm
  - Assist with coordinating development, integration of the common software
  - Address collaboration-wide needs and requirements not easily managed by individual experiments
    - Ensure interoperabilty
    - Manage evolution of the common architecture
    - Promote policies and tools to assist with development and integration
    - Carry out other actions as may be needed
  - Work closely with experiments to ensure alignment of goals, priorities and processes

## Projects and initiatives

- Directed at integration and collaboration-wide needs and requirements
- Recent and on-going projects and initiatives
  - Continuous integration system
  - Architecture review and revision project
  - Code profiling and optimization project
  - LArSoft / light-weight analysis framework (LArLite) integration project
  - Workshops and training
    - LArSoft Continuous Integration Workshop (June 2014)
    - LArSoft Architecture and Testing Workshop (June 2015)
    - art / LArSoft Course (Aug 2015)
    - LArSoft Requirements Workshop (Oct 2015)

## Continuous Integration System

- Continuous integration (CI)
  - A software development technique that involves frequent integration of newly developed code to the main branch of development
  - Each integration is tested by an automated build and test system
- Goals
  - Catch integration problems, unintended side-effects quickly
  - Maintain a more stable main-line development branch
  - Able to create releases with known properties at all times
- Deployed a CI system in September, 2014
  - Runs a developer-maintained test suite on every commit to the main development branch
  - Easy to configure, add tests to the suite
  - Can define tiers of test suites to be run at various times
    - Per commit, nightly, during release creation process

## Continuous Integration System

- Current status
  - Working on developing production quality, per-commit test suite
    - Runs unit tests
    - Runs all major components of the data and MC production chains for each experiment looking for major changes
    - Regression tests of known problems, data backwards compatibility
    - Executes and returns a result within about 10 minutes
    - Can be run on user code prior to committing to the main development branch
  - Will be followed by developing test suites to be run nightly, during release procedure.
    - Higher statistics tests, more in-depth probing for changes
  - Have 0.5 FTE working on this

## Architecture review and revision project

- Good design facilitates development of sophisticated algorithms
- Have promoted a set of design principles and objectives
  - Detector-independence using common interfaces to configuration data, generic indexing tools to automate loop construction, etc.
  - Factorization of algorithm code from framework interfaces
  - Modularization of algorithms into testable units with standardized interfaces
  - Accompanying tests to be run by the CI system
  - Documentation

## Architecture review and revision project

- Initiated a review of selected major components
  - Address problems in the areas described
  - Develop exemplars for the design principles
- Prioritization of the revisions
  - Interoperability
  - Factorization
  - Maintainability, modularization
  - Optimization
- Currently have about 0.5 FTE devoted to this effort
- This phase will be completed around end of calendar 2015.

# Code profiling and optimization project

- Optimization of algorithms can often benefit from advise, guidance from software engineering experts
  - Have enlisted a group from SCD to assist with profiling specific production chains to help identify major bottlenecks and suggest solutions
    Krzysztof Genser, Jim Kowalkowski, Hans Wenzel
  - A short project in winter 2015 focused on reconstruction
    - Report generated a number of tasks that uBooNE is working on
  - Existing project looking at the simulation, in particular:
    - Use of physics lists
    - Geometry use and energy deposition, stepping and other GEANT4 parameters
    - Code speed
    - Upstream detector integration
  - Allocated 0.5 FTE for about a month
  - Current project will conclude soon

## LArSoft / LArLite integration project

- uBooNE collaborators developed a light-weight development environment  – LArLite
  - Allows isolation of code elements to speed development cycle
  - Based on a different build system
  - Used by a number of uBooNE developers, particularly those working on shower reconstruction
- Several differences prevented easy integration with LArSoft
  - Data structures differed in some details
  - Incompatible data files required conversion from LArSoft to LArLite format

## LArSoft / LArLite integration project

- Initiated a project to integrate LArSoft and LArLite
  Dave Dagenhart, Chris Jones, Marc Paterno
  - Allow seamless, transparent migration of
    - data structures
    - algorithm code

    between LArSoft and LArLite
  - Ideally, want to be able to share source repositories for common code
- Initial report identified the necessary changes to LArLite, LArSoft and art framework
- Project work started earlier in the summer
  - Some of the required changes to LArLite already made
  - Working on changes to art framework
  - Much of the LArSoft work being carried out under the auspices of the architecture project

## Development and integration coordination

- A major issue for LArSoft collaboration to manage
  - Coordinating changes to common code driven by work of each experiment
  - Code developed by one experiment to address a specific issue can change behavior in undesirable ways for another experiment
  - Extremely important to understand, manage these types of situations
  - Need to do this while maintaining a stable development environment

## Development and integration coordination

- Have a number of tools to assist with coordination
  - Bi-weekly LArSoft Coordination Meeting
    - Attended by all experiments
    - All code changes are discussed, approved before being integrated into the main development branch
    - Contents of releases are discussed and approved
      - Special procedures for changes that break existing code
    - Discuss architecture, coding issues
  - Continuous integration system
    - An extremely important tool
    - Informs developers, offline managers immediately when changes adversely affect any experiment
  - Weekly integration releases
    - Frequent releases creates stable, common platform on which to base changes
    - All development occurs in branches separated from main development branch
    - Allows controlled integration procedure

## Development and integration coordination

- Have a number of tools to assist with coordination
  - Architecture Committee
    - Representatives from experiments plus SCD design experts
    - Explores architectural issues
    - Goal is to understand solutions, trade-offs prior to bringing them to the developers
  - Steering Group
    - Experiment spokes, Neutrino Division Head, SCD Division Head
    - Develops and approves priorities, policies and direction
    - Oversees meeting of requirements, milestones, etc

## Summary

- A diverse, very active community successfully contributing to LArSoft
- Core support team provides important services that support the collaboration
  - Have a good short term program of projects
  - Effort limited as to what we can do at present
- Still much more work to do
  - The present workshops are a good step toward defining how to proceed