# LArSoft code re-factoring and git transition status

R. Snider
*Fermilab*

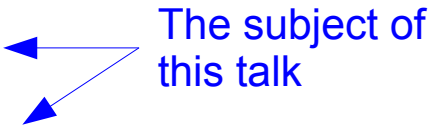
LBNE Software and Computing Meeting
Nov. 13, 2013

# LArSoft project

- LArSoft
  - An integrated, experiment-agnostic set of tools for the reconstruction and simulation of  LAr neutrino experiments, based on the art framework
- LArSoft project objectives
  - Exploit commonalities in detectors and technologies to
    - Leverage development effort
    - Reduce redundancy in software development
- Fermilab SCD responsibilities
  - Provide technical leadership, project management, infrastructure support, coordination for project
  - Coordinate and manage LArSoft releases and distribution
- Short term priorities
  - Address build, distribution, development environment diversity issues, and other tasks as requested by our user communities

# LArSoft project initiatives

- Transition to git + cmake from svn + SoftRelTools
- Re-factoring into experiment-specific and LArSoft core software
- Distribute LArSoft via cvmfs
  - Requested as short term goal by LBNE
  - Requested by uBooNE for grid compliance. Also a short term goal
- Off-site LArSoft builds
  - LBNE effort following on work on off-site art builds
  - uBooNE currently using off-site builds based on old system at some sites
- CPU / memory optimization
  - Particularly important for LBNE



# LArSoft project initiatives

- Pandora integration
  - A pattern recognition packaged developed for ILC studies
  - Requested by uBooNE for MC challenge this fall
- Port LArSoft to non-SLF platforms
  - Requested by uBooNE and LBNE
- Test framework
  - Project requirement

Communication, priority setting on project activities

- Attend LBNE and uBooNE software meetings + email lists
- Core project has weekly meetings with SCD, issues weekly status reports
- Convene “Stakeholder Meetings” with offline managers from experiments
- Convene “LArSoft General Meeting” for all users

Will return to regular meeting schedule after transition is completed

# Transition to git + cmake

- Goals
  - To improve the tools available to LArSoft code developers
  - To provide a more stable development environment
- The strategy
  - Migrate away from current svn repositories to git + git flow
    - Distributed model offers good isolation from head of “central” repository, while still providing fully featured VCS
    - Can share local work with others
    - Excellent merging capabilities to re-integrate with the “central” repository
    - Widely used (with growing popularity) within the IT industry
    - Use git flow to formalize, facilitate development workflow
  - Migrate away from current SoftRelTools build system to cmake
    - Provides a more modern, cross-experiment collaborative development environment
    - Has excellent support for parallel (multicore) and cross-platform builds
    - A simple, readable configuration language

# Status of transition

- The major steps
  - Factor code into git repositories
    - Chose 8 repositories, each of which contains packages at the same level of functionality
      - larcore, lardata, larevt, larreco, larsim, larana, lareventdisplay, larexamples
    - Dependencies between repositories is well understood, acyclic
    - Package each repository as a re-locatable ups product

    Status: done
  - Transition the build to cmake
    - Introduce necessary cmake configuration files
    - Take advantage of large cmake macro library and build toolkit used by art framework

    Status: done
    - Done
    - Created initial larsoft release based on this build infrastructure
    - Insufficient by itself to allow building of all LArSoft code

# Status of transition

- The major steps (cont'd)
  - Develop a tool to integrate the build across multiple repositories
    - “mrb” (multi-repository build system)
    - Functionality divided into two categories
      - Basic:  includes setup of environment, checking out packages, building, installing
      - Extended:  understands package dependencies, prevents incomplete builds
    - Require only the basic functionality for the first production release

    Status:  in progress

    - Created larsoft v0_00_04 for use in testing
    - Testing under way by E. Church, B. Rebel, R. Snider
      - Fixing bugs, finding places where improvements needed before beta release
      - Producing/testing basic end-user documentation
    - Expect beta release of basic functionality by the end of next week

# Status of transition

- The major steps (cont'd)
  - Create and validate beta release and build tools
    - Up-to-date code snapshot
    - Release to the experiments for testing
    - Test workflows, product functionality, documentation
    - Complete end-user documentation

    Status: pending completion of alpha mrb work
    - Will take a few days to create the beta release
    - Originally planned two weeks of testing with participation from stakeholder experiments
    - Tests conducted will form basis for test of first production LArSoft release

# Status of transition

- The major steps (cont'd)
  - Final production transition
    - Freeze svn, create a release
    - Perform final migration, create a release (functionally identical to svn version)
    - Validate by comparing before/after releases
    - Go-live

    Status:  pending beta release validation
    - Preparations to take less than a week
    - Final transition designed to proceed quickly to minimize downtime

    Overall, process is behind where we wanted to be at this time.
    Now anticipating completion in Dec.

# Re-factoring experiment and LArSoft code

- Currently all LAr software is in a single svn repository
  - Core detector-agnostic simulation, reconstruction + argoneut + LBNE + uBooNE + ...
- Goal of the re-factoring
  - To extract all experiment-specific code from core LArSoft
  - To enable experiment-specific build capability independent of LArSoft
    - Can schedule builds and releases independently of LArSoft
    - Experiment software will depend on LArSoft products
  - Retain detector-agnostic reconstruction, simulation algorithms in LArSoft
    - So far, highly successful at writing detector-agnostic code
    - Effort benefits from broad overlap of personnel across development communities

# Re-factoring experiment and LArSoft code

- Strategy
  - Introduce abstract interfaces in LArSoft for experiment-specific functionality
    - Ensure that each step in workflow is represented by module/interface in LArSoft
  - Implement interfaces in experiment repositories
  - Move non-code elements to experiment repositories
    - Configuration files (fcl)
    - Geometry description files (gdml)
  - Focus first on low-level packages
    - Geometry, data structures, low-level algorithms and utilities (larcore, lardata, larevt)
  - Target re-factoring work to proceed in parallel with git/cmake transition
    - Release together

# Status of re-factoring

- Survey of low-level packages nearly complete
  (links to results here: https://cdcvs.fnal.gov/redmine/projects/larsoft-experiment-alpha/wiki/Wiki)
  - Code couplings fall into two categories
    - Explicit detector-specific decisions in otherwise generic code
    - Stand-alone algorithms and modules used only for one experiment
  - To address the first
    - Delegate detector-specific decision to art “service interface”
    - Decision and relevant actions live in experiment service implementation
    - Example is ChannelMapAlg in Geometry
  - To address the second
    - Create module or interface to represent the workflow step or functionality
    - This case then reduces to the first
    - Example is signal shaping services

# Status of re-factoring

- Testing these changes in alpha LBNE repository
  - Started with Geometry
    - Code written to de-couple ChannelMapAlg creation, configuration from LArSoft
    - Testing integrated build with mrb. Expect this to be completed within the next day or two
  - Estimate the balance of the known issues to be completed in approximately one week.
    - This puts it on about the same schedule as mrb

# Status of re-factoring

- Points of contact for re-factoring work
  - Discussions at software meetings
  - Followed by initial surveys of packages with T. Junk and B. Rebel
    - Identified the initial cases to address and the main areas of concern
  - Experiment contacts to review changes and identify testers
    - LBNE:  T. Junk, B. Viren, Q. Li, M. Potekhin
    - uBooNE:  E. Church, H. Greenlee

    Have already found a few (uBooNE) people interested in testing

    Will need more for beta release

Will discuss details of all changes at up-coming LArSoft Stakeholders and General meetings

# Documentation

- LArSoft project documentation, status reports, task lists
  (might need Fermilab services account/password)
  - https://sharepoint.fnal.gov/project/LArSoft/SitePages/Home.aspx
- Alpha documentation, work notes
  - https://cdcvs.fnal.gov/redmine/projects/larsoft-alpha/wiki/
- Current LArSoft documentation
  - https://cdcvs.fnal.gov/redmine/projects/larsoftsvn/wiki
- Future site of LArSoft documentation after transition
  - https://cdcvs.fnal.gov/redmine/projects/larsoft/wiki

# Summary and comments

- Priority projects git+cmake transition and initial re-factoring of experiment and LArSoft code nearing completion
  - Build system based on that used for art framework
  - Will propose that mrb be used for integrated experiment repository / LArSoft build as well
- A set of important projects identified to follow these transitions
  - Work on cvmfs distribution is proceeding in parallel
  - Pandora integration proceeding in parallel
  - Remaining projects still to be scheduled
- Project hiring a dedicated developer
  - Expected to start work after new year
  - Will contribute in all areas of project work and certain software development projects