

# LArSoft and Future Framework Directions at Fermilab

Kyle Knoepfel<sup>1,\*</sup>

<sup>1</sup>Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois, United States

**Abstract.** The diversity of the scientific goals across HEP experiments necessitates unique bodies of software tailored for achieving particular physics results. The challenge, however, is to identify the software that must be unique, and the code that is unnecessarily duplicated, which results in wasted effort and inhibits code maintainability.

Fermilab has a history of supporting and developing software projects that are shared among HEP experiments. Fermilab's scientific computing division currently expends effort in maintaining and developing the LArSoft toolkit, used by liquid argon TPC experiments, as well as the event-processing framework technologies used by LArSoft, CMS, DUNE, and the majority of Fermilab-hosted experiments. As computing needs for DUNE and the HL-LHC become clearer, the computing models are being rethought. Presented here are Fermilab's plans for addressing the evolving software landscape as it relates to LArSoft and the data-processing frameworks, specifically as it relates to the DUNE experiment.

## 1 Preparing for DUNE's computing needs

One of the top priorities for Fermi National Accelerator Laboratory (Fermilab) is to support and enable the success of the DUNE experiment. This not only includes issues related to the physical construction, commissioning, and operations of the facility and detector, but also the physics and computing support required to yield scientific results.

After presenting a brief overview of the DUNE detector, we discuss aspects related to DUNE's computing needs, specifically the data-processing framework and the LArSoft software toolkit.

## 2 DUNE overview

The DUNE experiment's primary physics goals target [1]:

- Physics related to  $\nu_\mu$  and  $\bar{\nu}_\mu$  oscillations into their respective electron-neutrinos,
- Increased sensitivity to hypothetical nucleon decay processes, and
- Potential observation of  $\nu_e$  signal should a supernova occur.

Achieving these goals is accomplished through the Long-Baseline Neutrino Facility, which includes all infrastructure necessary for the production of (anti-)neutrinos and the housing of the detectors that will record the neutrino interactions. Two detectors will be situated

---

\*e-mail: [knoepfel@fnal.gov](mailto:knoepfel@fnal.gov)

**Table 1.** Expected readout windows and corresponding data rates for neutrino beam and supernova physics scenarios [1]. The readout windows for neutrino-beam running are separately specified for single-phase (5.4 ms) and dual-phase (8 ms) LAr technologies.

Physics source	Rate	Readout window	1 channel	1 APA	Full Module
Neutrino beam	0.83 Hz	5.4 ms (8 ms)	12 bits	41.5 MB	6.22 GB
Supernova burst	—	~100 sec	300 MB	750 GB	120 TB

800 miles apart to allow neutrinos the requisite distance to oscillate as they transit the earth. Located at Fermilab, the *near detector* is necessary for understanding the distribution of neutrino energies and intensities that are crucial in the analysis of signals detected at the *far detector*, which will be located at the Sanford Underground Research Laboratory in Lead, South Dakota.

As the description of the near detector is included elsewhere ([1]), we will focus on the far detector, which is designed to include up to four underground modules, each of which contains a 17 kiloton liquid Argon (LAr) time-projection chamber (TPC). Each module houses 150 anode plane assemblies (APAs), which detect the charge of drifting electrons, created through the ionization of LAr by charged particles.

A single APA (in the module configuration considered here) is composed of 2560 wires or channels, each of which are sampled every 500 ns with 12-bit precision. A full module readout per 500-ns sample therefore contains 0.55 MB of data, which is referred to as a *trigger record*. As shown in table 1, the readout windows depend on the physical processes being detected and also the LAr technology being used. The computing technologies needed to satisfy these requirements must therefore be able to handle a very wide range of data rates.

A timeline showing some of DUNE’s anticipated milestones and their tentative dates is presented in table 2. DUNE would like to use the same type of software environment (perhaps in prototype form) for ProtoDUNE-II as it would use for normal DUNE data processing.

### 3 DUNE framework requirements

One of the differences identified between collider and neutrino experiments is that whereas collider experiments naturally process data according to triggered *events*, neutrino experiments process data according to *regions of interest*, which is an area of the detector that may contain signatures of a physics process relevant to the goals of the experiment. A given DUNE trigger record may contain multiple regions of interest, each of which may need to be processed together or independently, depending on the reconstruction or analysis task.

**Table 2.** Approximate timeline (as of November 2019) for various milestones of the DUNE experiment. The terms “SP” and “DP” refer to single-phase and dual-phase LAr technologies, respectively. Only the software relevant to this talk is presented.

Approximate date	Milestone	Software used
Aug. 2018	ProtoDUNE I-SP	<i>art</i> , LArSoft
Aug. 2019	ProtoDUNE I-DP	
Late 2021/Early 2022	ProtoDUNE II	Framework that satisfies DUNE’s needs; LArSoft
Aug. 2024	Installation of Module 1	
Aug. 2025	Installation of Module 2	
2026/2027	Start of DUNE data-taking	

To complicate matters, the definition of a region of interest may depend on whether detector signatures should be grouped temporally or spatially. The minimal processing unit for DUNE is, therefore, rather fluid and may not be the same from one processing stage to the next, posing an awkward scenario for collider-based frameworks that support rigid processing hierarchies (e.g. run  $\supset$  subrun/luminosity block  $\supset$  event).

It is not yet clear if an existing reconstruction framework is able to (be adjusted to) naturally support such a data-processing model. The DUNE experiment, however, has instituted a task force whose charge is to define the requirements necessary for such a framework. Fermilab, in particular, is anxious to help find a solution that can meet DUNE's needs by ideally leveraging the experience and framework software developed for and by other experiments.

## 4 Future framework directions at Fermilab

DUNE's current reconstruction framework is the *art* framework [2], which was born in 2010 as a fork of the CMSSW framework [3]. Since that time, both *art* and CMSSW framework developments have proceeded according to the needs of the experiments either framework supports. However, sustainability and maintenance concerns have triggered discussions regarding whether more common tools can be developed.

As a result of a community-wide effort, the HEP Software Foundation released a white paper that stated [4]:

“In many areas it is recognised that different experiments could have adopted common solutions, reducing overall development effort and increasing robustness and functionality. That model of duplicated development is not sustainable. We must endeavour to achieve better coherence within HEP for future developments to build advanced, open-source projects that can be shared and supported in common.”

This sentiment has been adopted by Fermilab, and as a result, a future-frameworks initiative has been launched to determine in what ways framework functionalities (if not frameworks themselves) can be consolidated. As of March 2020, discussions are ongoing regarding the feasibility of such a consolidation.

## 5 LArSoft

In addition to a computing framework, the DUNE experiment makes use of LArSoft, which is primarily a C++ toolkit of experiment-agnostic LAr TPC reconstruction algorithms. The goal of the LArSoft collaboration is to provide common software tools that all LAr TPC experiments can use [5]. It also interfaces with other third-party libraries such as ROOT [6], Geant4 [7], WireCell [8], and PandoraPFA [9]. Many of its algorithms rely on *art* concepts and data structures.

Begun in 2008 by a postdoctoral associate, the LArSoft software project is now formally supported by Fermilab's scientific computing division. LArSoft is primarily supported by the contributions of graduate students, postdoctoral associates, and staff scientists from multiple experiments and many institutions. Most LArSoft developers are tasked with doing physics research with generally no funding for LArSoft development. That reality, as well as several others, leads to challenges that must be addressed.

## 5.1 LArSoft challenges

The LArSoft project has faced several challenges:

- It was created prior to C++11, and thus it has required noticeable effort to upgrade to newer language standards and coding paradigms.
- The current code base is somewhere around 350K lines of code, which is quite substantial for the number of library maintainers available.
- Most additions to the code base are made by C++ non-experts, who do not often have code maintainability in mind.
- Contributed code is rarely removed even if it is no longer used, creating a maintenance problem.
- The software release model has relied upon a single code librarian merging features by hand, without the use of automated tools for reviewing and approving code changes.
- Proposed breaking changes must be presented (in summarized form) at a coordination meeting, but no formal review process has been in place to carefully vet the changes.

As a means of addressing these challenges, the LArSoft collaboration moved to using GitHub and its pull-request model in January 2020. To effect the migration, many of CMSSW's GitHub scripts were ported over to work for LArSoft. Such a migration is expected to improve the collaboration model, bolstering the review level of each proposed code change.

The LArSoft collaboration has also supported a significant effort to clean up the code base. Specifically, many unnecessary files, functions, and virtual tables have been removed, and many header and library dependencies have been pruned to reduce compile times, library sizes, and runtime overhead. Another aspect of this effort is for LArSoft to reduce its coupling to the *art* framework, which helps to clarify the intention of the LArSoft toolkit, improve the separation of concerns, ease the maintainability of the code base, and improve the testability of the algorithms supported by LArSoft.

Lastly, as several LArSoft facilities were not designed with thread-safety in mind, Fermilab developers are re-architecting portions of the toolkit to be naturally thread-safe. Part of this effort is to educate LArSoft contributors in adopting thread-safe coding patterns. Such suggestions include (1) removing reliance on global, mutable state, (2) adopting “const all the things” coding idioms, and (3) favoring declarative coding patterns over procedural ones.

## 6 Conclusions

A chief goal of Fermilab is to enable the successful execution of the DUNE experiment, which includes helping meet its computing needs. Effort is underway to define the requirements necessary for DUNE's computing framework, and to improve the quality of the LArSoft product by adopting modern collaboration tools such as GitHub. Discussions are underway to determine how extant event-processing frameworks can be consolidated, and whether such consolidation can deliver a framework that meets DUNE's needs. Ultimately, the goal is for software to be ready by the time DUNE needs it.

The author would like to thank Dr. Christopher Jones (Fermilab) for delivering this paper in presentation form at CHEP 2019. He also thanks Dr. Michael Kirby (Fermilab) for helpful conversations and corrections regarding DUNE's computing needs.

---

## References

- [1] B. Abi *et al.* [DUNE Collaboration], arXiv:1807.10334 [physics.ins-det].
- [2] C. Green, J. Kowalkowski, M. Paterno, M. Fischler, L. Garren and Q. Lu, J. Phys. Conf. Ser. **396**, 022020 (2012).
- [3] F. Fabozzi, C. D. Jones, B. Hegner and L. Lista, IEEE Trans. Nucl. Sci. **55**, 3539 (2008).
- [4] J. Albrecht *et al.* [HEP Software Foundation Collaboration], Comput. Softw. Big Sci. **3**, no. 1, 7 (2019) [arXiv:1712.06982 [physics.comp-ph]].
- [5] E. L. Snider and G. Petrillo, J. Phys. Conf. Ser. **898**, no. 4, 042057 (2017).
- [6] R. Brun and F. Rademakers, Nucl. Instrum. Meth. A **389**, 81 (1997).
- [7] S. Agostinelli *et al.* [GEANT4 Collaboration], Nucl. Instrum. Meth. A **506**, 250 (2003).
- [8] M. Diwan, M. Potekhin, B. Viren, X. Qian and C. Zhang, J. Phys. Conf. Ser. **762**, no. 1, 012033 (2016).
- [9] J. S. Marshall and M. A. Thomson, J. Phys. Conf. Ser. **396**, 022034 (2012).