

LArSoft Architecture review

Phase I closing report

Gianluca Petrillo, Saba Sehrish

Fermi National Accelerator Laboratory

LArSoft Coordinators' Meeting, January 19th, 2016



Goals of the architecture review

- support experiment independence
- achieve framework independence
- increase maintainability
- promote testability

We identified five main areas for intervention:

- 1 **interoperability**: algorithms can operate independently from the specifics of the experiments
- 2 **factorization** of data, algorithms and services from the specificity of the environment
- 3 **generic interface** for common steps that need different implementations
- 4 **software architecture**: solution to design flaws that render developing LArSoft code unnecessary cumbersome
- 5 **maintainability**: code should be understandable and extensible with a moderate effort

Sub-project schedule

- set a way-point with the end of 2015 (end of LArSoft “phase I”)
- definition of the next goals will follow, including input from LArSoft requirements workshop

The area of intervention is vast:

- some effort was directly under the LArSoft Architecture sub-project, either
 - fully by the LArSoft project team
 - with main contribution from people from the Experiments
- some was in parallel with other sub-projects, e.g.
 - Continuous Integration system (for testing; lead V. Di Benedetto)
 - LArSoft/LArLite integration (iteroperability; lead C. Jones/M. Paterno)

- **extended Geometry service interface**
 - geometry IDs, uniquely identifying a cryostat, TPC, wire plane and wire, made easier to use:
 - sortable, printable, convertible and comparable one to another
 - all methods accept geometry ID as arguments
 - `IterateXxxx()` methods make iteration through all the elements of geometry easier
- **demonstration: ClusterCrawler module**
(thanks to B. Baller for support and patience)
 - was not compatible with multiple-TPC geometries: fixed
 - could not accommodate “disambiguation” phase between hit and cluster finding: redesigned in two separate modules
- **core services allow more experiment-specific specialization**
without losing generality
- **a few more modules were planned, but got de-scheduled**

Factorization

Major action on services:

- **most basic data products simplified** not to require framework support (**should have been *all***)
- **core services rely on a framework-independent service provider** — they will be **available in with LArSoft v5** (*thanks J. Paley*):
Geometry, LArPropertiesService,
DetectorPropertiesService, DetectorClocksService
- **new services designed with this separation** (*thanks B. Eberly*):
ChannelStatusService, DetPedestalService
- **demonstration:** ClusterCrawler
code split into algorithm and module
- **many new modules have been designed by the authors**
incorporating **this prescription**
- a separate project is lead by C. Jones and M. Paterno to interface LArSoft algorithms to MicroBooNE's LArLite

- a lot of discussion on how to implement generic interfaces...
- ... but no actual specific design
- spin-off project lead by S. Sehrish on review of `Track3DKalmanHit` will include the first implementation on track reconstruction
- interesting independent work by D. Adams to have TPC wire simulation modular, customizable and with a shared interface

Maintainability:

- event display was not redesigned
work in this direction will happen during this year
 - spin-off project on review of `Track3DKalmanHit` also focuses on maintainability
 - good practices preached and advertised
 - added numerous unit tests
-

Software architecture:

- performance analysis of MicroBooNE simulation by K. Knopfler and P. Russo, with recommendations
- no other specific work performed...
- thread-safety known to be an issue in some of the core services
- a lot of discussion is ongoing including *art* project on multi-threading

Summary

- the intervention area of the sub-project was vast
⇒ heavy prioritization and pruning was necessary
- **not all the areas could be addressed** as we would have liked
 - mainly for lack of time
- the most relevant achievements:
 - **increased authors' awareness on the topic**
(LArSoft school and workshop gave fundamental contribution)
→ **response has been very satisfying**
 - many data products are easily portable
 - core services now allow for fully factorized code
- key components of factorization are in LArSoft version 5
(today at [Release Candidate 2](#))
- two spin-off works (`Track3DKalmanHit` and `ShowerReco3D`)
are still in the process of being completed
- documentation will keep being integrated and improved

Next phase will be defined in the coming weeks — plenty of work to do!