Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

# *art* framework efforts

Kyle J. Knoepfel
HSF frameworks working group
29 January 2020

https://art.fnal.gov

## How do you spell *art* ?

- Many people spell *art* incorrectly.
  - It's okay…we brought it on ourselves.

| | | |
|---|---|---|
| ART | ✗ | Not an acronym, nor a preprocessor macro |
| Art | ✗ | Not this either |
| art | ✓ | This is acceptable and easiest to type |
| *art* | ✓ | This is preferred (with italics) |

- Honestly, I don't care too much.  But now you know. ☺

Fermilab

## *art* concepts

- Hierarchical data processing ($run \supset subrun \supset event$)
- **Experiments decide** how to define the processing levels (e.g. event)
- All processing elements are plugins, loaded at run-time via user configuration
  - Input source
  - Data-processing modules
  - Output modules
  - Other utilities that facilitate data-processing
- *art* provides various input sources and output modules, but all processing elements can be user-defined
- Workflows are assembled by a configuration file loaded at run-time
  - Adjustments to workflows do not require recompilation of C++ source code

Fermilab

## Highlighted features

- **Core framework behavior**
  - Concurrent processing of events supported within a subrun (inspired by CMS)
  - Data-product management is thread-, type-, and `const`-safe
  - Core framework functionality does not depend on ROOT
    - We support a separate package (art-root-io) that provides a ROOT I/O layer
  - Output file rollover based on user-defined criteria (e.g. max. events processed)
  - Implicit data-product aggregation for non-event products
  - Secondary input (backing) files

## Highlighted features

- **Core framework behavior**
  - Concurrent processing of events supported within a subrun (inspired by CMS)
  - Data-product management is thread-, type-, and `const`-safe
  - Core framework functionality does not depend on ROOT
    - We support a separate package (art-root-io) that provides a ROOT I/O layer
  - Output file rollover based on user-defined criteria (e.g. max. events processed)
  - Implicit data-product aggregation for non-event products
  - Secondary input (backing) files
- **Usability features**
  - Configuration description and validation suite
  - Module time- and memory-tracking facilities
  - Graph of data dependencies between modules

## *art* software products

- The *art* project distributes several software products.

Full-feature framework and its underlying packages.

## *art* software products

- The *art* project distributes several software products.

Full-feature framework and its underlying packages.

Lightweight library that allows reading of *art*/ROOT files; does not create new data products.

Ported from CMSSW's FWLite by a CMS developer.

## *art* software products

- The *art* project distributes several software products.

art

Full-feature framework and its underlying packages.

art GALLERY

Lightweight library that allows reading of *art*/ROOT files; does not create new data products.

Ported from CMSSW's FWLite by a CMS developer.

art STUDIO

Lightweight package that provides a development sandbox for testing new user-defined *art* modules.

## *art* provides the framework needs for ~2k physicists

ArgoNeuT

*artdaq* project

DUNE
DEEP UNDERGROUND NEUTRINO EXPERIMENT

Imaging Cosmic And Rare Underground Signals

LArIAT experiment

LArSoft

μBooNE

Mu2e

μ g−2

NOνA

SHORT-BASELINE SBND NEAR DETECTOR

Previous and potential users

DARKSIDE

next

supernemo collaboration

## *art*'s development approach

- Our development efforts are guided by:
  - Current and future needs of *art*-using experiments
  - Current and future software and hardware technology, in and outside of HEP
  - Feedback from individual users, and our own estimation how to improve *art*'s usability use
  - Laboratory priorities

## *art*'s development approach

- Our development efforts are guided by:
  - Current and future needs of *art*-using experiments
  - Current and future software and hardware technology, in and outside of HEP
  - Feedback from individual users, and our own estimation how to improve *art*'s usability use
  - Laboratory priorities
- Experiment support
  - Design guidance and code reviews at the request of experiments
  - Small-scale profiling efforts at the request of experiments

## *art*'s development approach

- Our development efforts are guided by:
  - Current and future needs of *art*-using experiments
  - Current and future software and hardware technology, in and outside of HEP
  - Feedback from individual users, and our own estimation how to improve *art*'s usability use
  - Laboratory priorities
- Experiment support
  - Design guidance and code reviews at the request of experiments
  - Small-scale profiling efforts at the request of experiments
- Dedicated biweekly stakeholder meetings
  - Discussion of upcoming changes and issues with stakeholders
  - Sharing among experiments

## Distinctive aspects of the *art* user community

- Many *art* users are doing development for experiments that are not yet running:
  - Reconstruction algorithms not yet finalized
  - Workflows are under development
- They are often involved in multiple experiments at the same time.
- They are involved in software development including event-generation, material simulation, processing raw data, reconstruction, to analyzing quantities of physical interest.
- They are defining experiment-specific data models.
- They are generally willing to (drastically) rethink any stage of the physics workflow:
  - Event representation, exploring other I/O libraries, etc.

## Distinctive aspects of the *art* wrt other frameworks

- There is clear boundary between framework code and experiment code
  - Helpful conceptual distinction for users
  - Makes R&D easier

## Distinctive aspects of the *art* wrt other frameworks

- There is clear boundary between framework code and experiment code
  - Helpful conceptual distinction for users
  - Makes R&D easier
- Potential for experiments to want conflicting framework behaviors/features
  - True in principle; but it has not happened in 10 years

Fermilab

## Distinctive aspects of the *art* wrt other frameworks

- There is clear boundary between framework code and experiment code
  - Helpful conceptual distinction for users
  - Makes R&D easier
- Potential for experiments to want conflicting framework behaviors/features
  - True in principle; but it has not happened in 10 years
- Why has it never happened?
  - We strive hard for stakeholder consensus on any given feature.
  - We provide enough flexibility in the framework that each experiment's needs can be met.
  - We are very conscious of the XY problem.
  - We take software engineering very seriously—any feature request that we think is unmaintainable down the road is not implemented.
  - We will not implement a feature that conflicts with the *art* processing model.

## Limitations of *art*

- The atomic processing unit for DUNE is fluid.
  - DUNE thinks in terms of a trigger record, which may contain multiple regions of interest.
  - Each region could be considered an “event,” but the shape of the event is not necessarily consistent from one processing stage to the next.
  - Instead of rigid hierarchy of Run ⊃ SubRun/Luminosity Block ⊃ Event, the hierarchy might be *user-defined <u>and dynamic</u>*.

- Collider-based frameworks are an awkward fit.
  - Can one of the existing frameworks be adjusted to support such processing?
  - Is a new framework required?

## Limitations of *art*

- The atomic processing unit for DUNE is fluid.
  - DUNE thinks in terms of a trigger record, which may contain multiple regions of interest.
  - Each region could be considered an "event," but the shape of the event is not necessarily consistent from one processing stage to the next.
  - Instead of rigid hierarchy of Run $\supset$ SubRun/Luminosity Block $\supset$ Event, the hierarchy might be *user-defined <u>and dynamic</u>*.

- Collider-based frameworks are an awkward fit.
  - Can one of the existing frameworks be adjusted to support such processing?
  - Is a new framework required?

Fermilab's future frameworks initiative (FFI) seeks to answer these questions.

## FFI item 1: A convergence of two frameworks?

> *"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."*
>
> \- The HEP Software Foundation, Albrecht, J., Alves, A.A. *et al*. Comput Softw Big Sci (2019) 3:7

- *art* was born in 2010 as a fork of the CMSSW framework.
- Since that time, both *art* and CMSSW framework developments have proceeded according to the needs of the experiments each framework supports.

## FFI item 1: A convergence of two frameworks?

> *"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."*
>
> - The HEP Software Foundation, Albrecht, J., Alves, A.A. *et al*. Comput Softw Big Sci (2019) 3:7

- *art* was born in 2010 as a fork of the CMSSW framework.
- Since that time, both *art* and CMSSW framework developments have proceeded according to the needs of the experiments each framework supports.
- Sustainability and maintenance concerns have triggered discussions regarding the feasibility of consolidating the *art* and CMSSW frameworks into one.
- Discussions are ongoing.
- **Bottomline:** Fermilab takes the HSF-mentioned concerns very seriously.

## Current efforts

- *art* development has stalled until Fermilab's framework plans are more concrete
- We are working hard to help users benefit from *art*'s multi-threading support
  - Lot of effort toward upgrading LArSoft, a Fermilab-supported liquid Argon software toolkit
- We are moving toward a Spack model of delivering software
  - Designed for HPC systems
- We are working with some experiments/projects in using HPC systems
  - Includes use of alternative I/O systems (e.g. HDF5, object stores)
- Many more things, too…

## Current efforts

- *art* development has stalled until Fermilab's framework plans are more concrete
- We are working hard to help users benefit from *art*'s multi-threading support
  - Lot of effort toward upgrading LArSoft, a Fermilab-supported liquid Argon software toolkit
- We are moving toward a Spack model of delivering software
  - Designed for HPC systems
- We are working with some experiments/projects in using HPC systems
  - Includes use of alternative I/O systems (e.g. HDF5, object stores)
- Many more things, too…

*Thank you*