What is a framework, and what is it for?

Kyle J. Knoepfel

Framework definition

- The term “framework” is used often in HEP, usually without definition. The assumption is that “you know it when you see it,” or that you recognize the contexts in which it is used (HLT, reconstruction, etc.).
- A decent definition from Wikipedia (https://en.wikipedia.org/wiki/Software_framework):

> In computer programming, a **software framework** is an abstraction in which software, providing generic functionality, can be selectively changed by additional user-written code, thus providing application-specific software.

Framework definition

- The term “framework” is used often in HEP, usually without definition. The assumption is that “you know it when you see it,” or that you recognize the contexts in which it is used (HLT, reconstruction, etc.).
- A decent definition from Wikipedia (https://en.wikipedia.org/wiki/Software_framework):

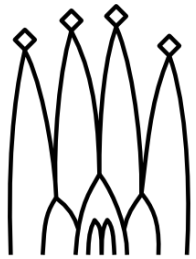
> In computer programming, a **software framework** is an abstraction in which software, providing generic functionality, can be selectively changed by additional user-written code, thus providing application-specific software.

- The user “plugs in” their code to a framework, often through dynamically loaded libraries called “plugins”.
- In (e.g.) art and CMSSW, this means that the framework owns the ‘`int main(…)`’ function, which calls user code under the covers.
- Frameworks are often used in a high-level trigger environment, for reconstructing physics objects from detector signals, or for simulating physics processes.
  - HEP frameworks are often not used in the context of analysis.

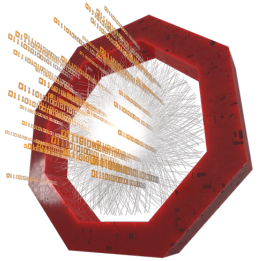
Framework definition (in pictures)

Framework definition (in pictures)

The framework is responsible for **driving** the offline data loop.



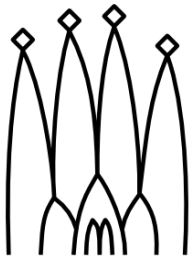
Gaudi



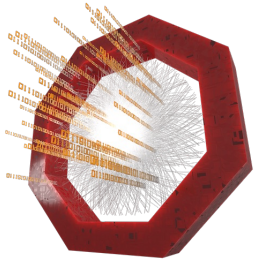
Alice's O2

Framework definition (in pictures)

The framework is responsible for **driving** the offline data loop.



Gaudi



Alice's O2

Physics algorithms **plug-in** to the framework to process each piece of data provided by the loop.

e.g. LArSoft

Fermilab DUNE

Existing frameworks

There are several existing frameworks:

- ***art***: Used by most intensity-frontier experiments at Fermilab. Originated as a fork of CMSSW ca. 2010.
- **CMSSW framework**: Used by the CMS experiment. CMSSW comprises the framework, other core functionality, physics algorithms, etc.
- **Gaudi**: Used by ATLAS, LHCb, Daya Bay, and MINERvA. Gaudi comes in different flavors highly tailored according to the experiments' needs.
- **JANA2**: Used by the Electron-Ion Collider.
- **O2**: Used by ALICE.
- etc.

What we are *not* talking about

Examples that are not frameworks wrt. this task force:

- **LArSoft**: LAr toolkit/library that is designed to be framework-agnostic
- **ROOT**: toolkit used ubiquitously in HEP analysis and as underlying library for persisting C++ data structures to disk.
- **Geant4**: library used for detector simulation, response, etc.
- **A particular computing language (C++, Python, etc.)**: Even though a framework is often implemented in a specific language, the language is not a framework.
- **Small programs written by physicists**: often used for individual analyses.

What we are *not* talking about

Examples that are not frameworks wrt.

- **LArSoft**: LAr toolkit/library that is designed to be framework-agnostic
- **ROOT**: toolkit used ubiquitously in HEP analysis and as underlying library for persisting C++ data structures to disk.
- **Geant4**: library used for detector simulation, response, etc.
- **A particular computing language (C++, Python, etc.)**: Even though a framework is often implemented in a specific language, the language is not a framework.
- **Small programs written by physicists**: often used for individual analyses.

A framework is intended to be used among collaborators.
It is a means of sharing code, data, and workflow patterns.

A framework should allow you to…

- Work on physics algorithms without you needing to know the details of how data are passed between them.

- Specify your own data model
  - What data objects do you care about?
  - How do they relate to each other?
  - In what data organization layer do they live (event, subrun, etc.)?

- Assemble and execute a sequence (or graph) of algorithms based on a simple specification (e.g. configuration)
  - The framework should figure out how to piece it all together.

A framework should provide…

- An I/O layer so you don't have to explicitly write/read framework data to/from disk
- A system that records the provenance of the produced data
  - Includes storing the framework job configurations and tracking data product parentage
- Composable workflows, allowing separate processing stages
- Centralized support for:
  - Configuration access
  - Random number management
  - Multi-threaded processing
  - Message logging
  - Access to (or at least heuristics to access) entities outside of the framework
- Utilities that help you improve the way you process your data
  - Basic profiling tools

Example: **What does *art* do?**

- **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 (e.g. automatic combination of protons-on-target counts)
  - Secondary input (backing) files

- **Usability features**
  - Configuration description and validation suite
  - Module time- and memory-tracking facilities
  - Graph of data dependencies between modules

Lessons learned

- *art*'s rigid processing hierarchy has been an awkward fit for neutrino experiments
  - We are pursuing something more flexible (https://github.com/knoepfel/meld)

- *art* supports a class a plugins (i.e. services) that can be accessed from anywhere.
  - This has led to many thread-safety issues that experiments must deal with. We should steer clear of this.

- *art* users can access metadata and provenance about data products
  - Many users do not look at this information
  - Metadata and provenance are important when you need it, but we should find a simpler way to provide it.

- Framework limitations are not necessarily bad! But you should know what they are.

Other references

- Talk by Adam Lyon at Fermilab Frameworks Workshop in June
  - https://indico.fnal.gov/event/59872/contributions/267310/attachments/167641/223737/framework_workshop_2023_lyon.pdf

- art.fnal.gov (somewhat out of date)

- HSF frameworks working group
  - https://hepsoftwarefoundation.org/workinggroups/frameworks.html