# Software Framework — LAr

ND Software Integration Meeting

April 30, 2020

Andy Mastbaum

Rutgers University

mastbaum@physics.rutgers.edu

# SFRT Requirements

- **Discussed the DUNE SFRT Writeup Draft in the LAr group**
- **Long-term framework needs are hard to anticipate as LAr ND software is still in development; trying to look ahead based on current status/plans**
- **Overall, not many comments — just a few points to highlight:**

- The data files should be stored in a format such that they are readable without requiring access to the experiment framework code.
- The framework should support straightforward interfacing to external library code, to enable delegation of arbitrary simulation, analysis, and processing steps to external packages.
- The data format should enable batch processing of chunks of data, including fast conversion to array for staged loading onto GPUs with modest memory resources.
- Provenance metadata is required such that each event in a file can be reproduced (re-simulated or re-processed under identical conditions, with identical results). This does not necessarily require provenance information to be stored event-by-event.

**continued...**

# SFRT Requirements

- The framework should support parallel, distributed (e.g. MPI over servers) workflows that are highly configurable (e.g. rank optimization), including asynchronous processing of arbitrary subsets of arbitrary groups of events by generic computing devices (e.g. CPU parallelization, GPUs, FPGA, client-server "analysis as a service" architectures).
- The framework requirements should avoid tying to any specific implementation of file format used to achieve (parallel) I/O requirements, e.g. ROOT friend trees. This could even include using "files" at all, as opposed to a form of database.
- The framework should be agnostic to the format of the raw data, i.e. waveforms vs. hit-like as in a pixel readout. However, high-level interfaces should ideally enable the equivalent derived objects (e.g. reconstructed 3D space points) to be used interchangeably in analysis code.
- Relatedly, the framework should make no assumptions about the detector geometry or sensor types (e.g. requiring a wire-based LArTPC).
- It should be possible to perform stages iteratively, to produce new outputs. Example: running "stage 1" reconstruction independently in the LAr ND and MPD, then "stage 2" using those outputs to perform a cross-detector reconstruction. This implies that framework output can become input.

**Google Doc for additional comments**

# Data Model

ND Software Integration Meeting

April 30, 2020

Andy Mastbaum
Rutgers University
mastbaum@physics.rutgers.edu

# Data Model

- **ND Software Data Model**
  - As discussed previously, it is important to settle on a robust data model for simulation & analysis as soon as possible
  - "Data Model" refers to the information stored; there is a related but separate question of the actual file format (`TTrees`, etc.)
  - Seeking consistency as possible across the ND: common analysis tools, enabling joint high-level reconstruction (matching, etc.)
    - Proposal to centralize through G4 simulation on gevgen_fnal + edep-sim provides a step in this direction
- Consistency with the FD data model is also desirable
  - This has extensive documentation, useful for a comparison and a reference point for an ND-specific document
- Input from all ND groups and analyses will be crucial!

# Data Model

Data Model Taskforce



## DUNE DocDB 14392

DUNE Data Model Taskforce Document

- Data flow
- Raw data format
  - Configuration
  - Translation to offline format
  - Merging
- Metadata
- Offline format
  - Processing stages
  - Full & reduced/skim files
  - Analysis files

# Data Model Committee

- **A ND Software Integration subcommittee on Data Model**
  - Following discussions with ND software conveners, we believe it would be helpful to form a new *ND Software Integration subcommittee on Data Model*
  - Seeking members and input, look out for announcements
    - Representation from ND detectors, SW integration, and ideally the FD data model task force.
- Initial steps: "Diff" of current detector data model plans
- Outcome: A document describing a common ND data model, including where commonalities and differences are across the ND (and FD)
- We'll want to get this in place ASAP, to avoid divergence as software development continues — will call dedicated meetings soon
- Your input will be essential to putting this together!

# Data Model Committee

**LAR ND Simulation Data Products**
A. Mastbaum, 2020/03/25
Based on the concept from D. Dwyer, F. Piastra, B. Russell, and K. Terao, see slides

- Generator → Truth events
  - Generator ID (e.g. GENIE, NuWro, CORSIKA, particle gun …) [enum]
  - Version [string]
  - Configuration/tune [string]
  - Event ID (run, subrun, spill) [struct]
  - Interaction ID [unsigned]
  - GHepRecord
    - [genie::GHepRecord or equivalent]
  - Flux record
    - [bsim::Dk2Nu or equivalent]
  - Spill record
    - Beam conditions, spill intensity (TBD with beam group)
  - Extra generator information (generic container)

- Tracking (edep-sim G4) → True interaction steps
  - Event ID (run, subrun, spill) [struct]
  - Interaction ID [unsigned]
  - Primary particles [TG4PrimaryVertexContainer]
  - Trajectories [TG4TrajectoryContainer]
    - Initial/final position/momentum four-vectors
    - Physics process
  - Hit detector segments [TG4HitSegmentDetectors]

- Signal Propagation → True hits
  - Event ID (run, subrun, spill) [struct]
  - Charge hit collection
    - Channel ID (pixel) [unsigned/struct]
    - Interaction ID [unsigned]
    - Track ID [unsigned]
    - Track step [unsigned]
    - Edep [float]
    - Time [float]
    - Landau fluctuation [float]
    - Recombination factor [float]
    - Longitudinal diffusion factor [float]
    - Transverse diffusion factor [float]
  - Optical hit collection
    - Detector type (ArcLight/LCM) [enum]
    - Channel ID (optical detector) [unsigned]

**MPD Data Products**
E. Brianne, 2020/03/27
Work from GArSoft Team: E. Brianne, L. Bellantoni, T. Junk, T. Mohayai and al.

- Metadata (Provenance) to be added, need to think how to do that: SAM, into the file...
  - Geometry ID(s)
  - Cluster node, software versions, etc.
  - Index into conditions DB/spreadsheet

- Generator → Truth events
  - Standalone storage
    - GHepRecord
      - [genie::GHepRecord or equivalent]
    - Flux record
      - [bsim::Dk2Nu or equivalent]
  - Spill record (TBD)
  - Storage in art
    - simb::MCTruth, simb::MCFlux, simb::GTruth (see nutools)

- Tracking/Geant4 simulation (edep-sim) → units in MeV, mm, ns
  - Event ID (run, subrun, spill)
  - Geometry ID + version (or full geometry)
  - Primary particles [TG4PrimaryVertexContainer]
  - Trajectories [TG4TrajectoryContainer]
    - Initial/final position/momentum four-vectors
    - Physics process
  - Hits in detector segments [TG4HitSegmentDetectors]
    - *Modifications needed in edep-sim? Birks' Law?*
    - *Implementation of the B-field map to be done (important for LAr also for fringe effects)*
  - Module written in art to convert edep-sim data format to art (see here)

- Tracking/Geant4 simulation in GArSoft (GArG4) → units in GeV, cm, ns
  - Event ID (run, subrun, spill)
  - Geometry ID + version (or full geometry)
  - MCParticles (based on nutools structure)
    - Particles created in G4 (decays, interactions…) + FSI (original mother)
  - Custom G4 Action to get hits
    - Energy deposits (TPC)
      - TrackID [int]
      - Time [float]
      - Energy [float]
      - Position [float, float, float]

Documentation of current/planned LAr + MPD Data Products
https://indico.fnal.gov/event/23883/