Thursday 22nd October, 2015

10:32

Requirements for a LArTPC Ecosystem

Lots of names here

Revision 0.1

Contents

1 Introduction

The purpose of this document is to capture what we, as the LArTPC community, want to provide to meet the physics requirements of our present and future liquid argon-based experiments.

The scope of these requirements is the data simulation, reconstruction and analysis eco-system necessary to extract physics results from Liquid Argon Time Projection Chamber (LArTPC) detectors used by experiments.

This eco-system includes the human interactions and responsibilities, software, computing hardware, workflows, processes, interfaces and communications, needed to deliver validated physics results.

The requirements will be used to guide appropriate development (including R&D) and operation of software and computing systems, personnel processes and interactions, testing and validation activities - with especial attention to those components that can be in common and/or shared.

The systems that are being, and will in the future be architected, designed, implemented and deployed are ultimately the experiment data simulation, reconstruction and analysis systems themselves specific to each collaboration.

1.1 Overview

The overall organization of the requirements and the gathering process should be discussed here. This is a reasonable spot to put a simple high-level diagram of the eco-system.

Requirements are organized by topic, established to facilitate the collection process. Topics are listed below.

- Non-beam reconstruction and analysis, including cosmic ray removal and non-beam particle identification.
- Beam reconstruction and analysis
- Overarching Analysis strategies, light detection systems and external detectors
- Human interactions, computing systems, software and interfaces.

1.2 Requirements organization

Requirements are grouped by topic. Within each topic requirement categories of functional and nonfunctional are defined.

User-level or User Requirements are the primary requirement type captured in this document.

Functional requirements are broken down by area of interest (as subsections).

Non-functional requirements include

- Performance Requirements,
- Interface and Protocol Requirements,
- Operational and Verification Requirements,

Note that any discussion of system-level requirements and design constraints that help to provide context will be in an appendix.

1.2.1 Requirement Attributes

The complete set of requirement attributes used in a formal process are listed below.

For this document we make judicious use of the set, as useful and relevant. Thus not every requirement will list every attribute. In many cases attributes will apply to several requirements collectively and will be described as such.

- ID - short string label that defines what requirement this is, including the area it is under.
- Description - short text that describes the requirement. Terms used should be from glossary.
- Parent ID - requirements that caused this requirement to exist.
- Requirement type - Science (Sc), Software (Sw), System (Sy), Human (H), Interface (I), Business (B).
- Priority - high, medium, low, wish
- Value goal - number or range
- Units - the units of the value goal
- Author(s) - Person(s) inserting or modifying this requirement
- Verification method - Data Analysis, Demonstration, Inspection, Simulation, Test/Measure...
- Time needed - <6 months; 6 months to 24 months; >24 months...
- Status - Proposed, accepted, validated, obsolete, superceded
- External reference(s) - background or other material relevant for this requirement.
- Comments - Any additional information that will help others to best understand this requirement, including examples from what currently exists.

A requirement should have a name and a clear description. In addition, it ought to have a short paragraph that describes attributes of the requirement: its importance, why it is here, what it is related to at a higher level, and how it will be validated.

1.2.2 Roles

The roles interacting with all systems described herein are listed below. There are primary and secondary roles. The primary ones will be the main focus for the requirements that will be collected.

Primary roles are:

- *Algorithm Developer* (AD). A software developer (scientist, computer scientist, computing professional) with specific expertise in developing and implementing algorithms for processing LArTPC data.
- *Algorithm tester / validator* (AT). Someone who makes sure the algorithms are working correctly and fit well into the system (performance, conformance, maintenance for example).
- *Scientist Data Analyst* (SDA). A scientist who uses the systems to perform scientific analyses of data.

Secondary roles are:

- *External Package Provider* (EPP). Provider of an external package that interfaces to and depends on or is depended upon by the LArTPC software solutions.
- *Infrastructure Developer* (ID). A developer of the computing, system and framework infrastructure.
- *Overseer* (O). Individual who is sponsoring, reviewing, reporting on the system's functionality and status.
- *Production Processing Operator* (PPO). Individual who runs the reconstruction systems for production data processing.
- *Scientist Configuration Developer* (SCD). Person that designs and consults on the configurations that would be utilized by the operators and users

2 Stakeholders

The stakeholders are those people and groups who will use, contribute to, support and/or oversee the full life-cycle of the LArTPC eco-system. Stakeholders include:

1. The experiments and project groups:
    - LArIAT
    - μBooNE
    - Icarus
    - SBND
    - SBN
    - DUNE-35t
    - Proto-DUNE
    - WA105
    - DUNE

2. Software and computing projects and toolkits developing, operating, supporting eco-system components:
    - *Nutools*
    - *PANDORA*
    - *Geant4*
    - *LArSoft*
    - *art*
    - *artdaq*
    - *FIFE*
    - *artG4*
    - Wire-Cell
3. Management and oversight of the above groups

3 Non-beam reconstruction and analysis

3.1 Description

Non-beam reconstruction and analysis includes removal (cosmic ray and radiologicals) and non-beam particle identification:

- Reconstruction, analysis and simulation
- Systematics and constraints
- Cosmic Ray Removal - from LArTPC alone, combining information from other detectors
- Scintillation light efficiencies
- Cosmic Ray analysis - for those experiments doing this
- Cosmogenic background sources for proton decay and supernova physics
- If can do astrophysical pointing - can do solar cosmic rays, moon shadow analyses
- Cosmic ray air-shower analyses with multiple neutrino detectors (need to synchronize detectors)
- Supernova with near detector?
- Simulate data starting from a generator; or generate by the simulation program

Calibrations...

- Calibrations (laser) - e.g. Calibration UV laser inside detector - for ionization.
- Movable (and changeable) radioactive source within detector
- View pulse shapes
- Inject signals into electronics
- Non-physics data (e.g. calibrating asynchronous clocks)

We identified the following use cases which primarily are the physics goals for the full DUNE detector:

- Detect and trigger on Supernovae.
- Search for nucleon decay
- Detect Atmospheric Neutrinos.
- Detect Solar Neutrinos
- Search for non-beam Exotics, for example Indirect Dark Matter production.

The following use-cases are tasks and capabilities which we felt are necessary to achieve the above goals, but not necessarily unique to these goals:

- Take data continuously.
- Process External Triggers.
- Handle Cosmics and radiological backgrounds.
- Reconstruct data from external detectors.
- Develop algorithms for individual sub-detectors independently.

3.2 Requirements

Calibration

- Data products for calibration tracks (e.g. additional data such as charge injection)
- Separate calibration and data runs (in metadata discoverable by data management)
- Calibration constants stored in conditions database and accessible by reconstruction and analyses

We identified the following requirements for the offline software

Online Software

- Ability to to trigger online for non-beam signals. (Trigger is meant to be an online condition that is looking at the data on a timescale short compared to the time between beam but longer than the timescale at which you are taking data, in order to recognize and record events of interest (supernova, proton decay, etc...))

Offline Requirements

Event Pick for supernova is the ability to grab events based on a “time-stamp” to search for supernova neutrino events based on a external alarm (SNEWS) (may be used differently elsewhere)

- Ability to read the online data format.
- Ability to split (in time/region), stitch, mix/overlay data and MC events. A wide variety of use cases, not necessarily related to non-beam physics, demand this requirement. For example using recorded backgrounds/noise from detector in simulation, removing cosmic or radiological backgrounds, or restricting reconstruction to a small region of the detector. (Non-Essential requirement)
- Efficiently overlay very large raw data events on GRID (such that separated jobs don’t repeat the same overlay?) in a statistically appropriate way, for example to mix recorded backgrounds/noise into simulation. (Non-Essential requirement)
- Event Pick facility to, for example, to navigate to arbitrary time-window in the data stream associated to a supernova.
- Handling (matching detector readout that may have different time scales and timestamps) of asynchronous data, for example for multiple sub-detectors and possibly across multiple detectors (e.g. SBN program and CAPTAIN-MINERVA).
- Perform non-LArTPC reconstruction. For example the Near detector for DUNE and auxillary detectors on most experiments.
- Ability to simulate and reconstruct sub-detectors independently.

- Interface to External Event Generators: For identified example cosmic ray generators (e.g. CRY, Corsika, Fluka) and relevant Supernovae (e.g. ...), atmospheric neutrino (e.g. Honda), proton decay (e.g. GENIE), and radiological generators. *We are unsure which of these two interpretations is meant by "interface" Interface with event generators and have the ability to run multiple simulation packages Interface with the output of external event generators*
- Generate individual events or collection of events in time. (We found this wording unclear so we produced the subsequent bullet point....not meant as a duplication or a separate bullet point)
- *Generate individual or collection of events with different time structures but that appear within a single readout of the entire detector (including sub-detectors)*
- Perform simulation, reconstruction, and analysis that is time dependent (seasonal/-solar cycle).
- Use and convert to Astronomic coordinates (zenith and azimuthal angles).
- Generate MC with data conditions.
- Ability to apply and decode flexible zero-suppression and lossless and lossy compression.
- Ability to extract physics from bad/calibration events. (does this mean the ability to do reconstruction even when things aren't working perfectly?)
- Desire: Flexible making of ntuples to serve different analyses.
- Timestamp events (at resolution of detector timing) for synchronization across detectors and matching (e.g. cosmic ray air show detector)
- Outputs from detector simulation programs have a standard format. And also need common geometry inputs – or master config (see below).
- Have a master configuration that gets translated to the inputs of the particular external packages.
- Generator level filtering - or filter the generator output. But may need reconstruction to do that filtering. E.g. running Genie filtering on some final state.

3.3 Cosmic Ray Identification and Removal

- Seems that the requirements for non-beam physics (once you've identified the window of interest in the data) is the same as for beam-physics cosmic ray removal and identification (e.g. Tagging, subtracting, simulating, etc...)
- Having the ability to put together from a continuous data stream an "event" from asynchronous readout when you do not have a fixed t = 0 (such as a beam trigger)

3.4 Removing Radon

Match PMT flashes to charge in TPC to see decays of Radon. Nanosecond scale timing on the light system with necessary light coverage. This drives:

- How fast you digitize
- How much data is produced

3.5 Stuff

Need the ability to see and reconstruct ¡ 10 MeV stuff.

Do reconstruction in passes - first pass identifies and tags regions of interest, subsequent passes focus on those regions separately (and focusing on regions of non-interest too).

3.6 Non-functional requirements

3.6.1 P

erformance requirements

- For non-triggering supernova detection, the ability to buffer a sufficient amount of data such that the experiment can respond to an external supernova trigger and retrieve the relevant data
- For triggering supernova detection, the ability to form a supernova trigger fast enough and with low enough fake rate to contribute to online supernova trigger (e.g. SNEWS)
- The ability to buffer and then readout all of the LArTPC data if an online proton decay trigger is satisfied
- The ability to readout “out-of-time” data from a beam trigger (for exotic searches)

3.6.2 I

nterface and protocol requirements

3.6.3 O

perational and verification requirements

3.7 Validation

3.8 Use Case Story

If you do a cosmic ray analysis with the current LArSoft and you want to produce a x10 amount of Monte Carlo sample compared to your cosmic ray data sample (which can be 100k+ events) the current size of the MC file makes this impossible.

4 Beam reconstruction and analysis

4.1 Description

Beam reconstruction and analysis includes:

- Reconstruction, analysis and simulation
- Systematics and constraints
- Beam particle identification - including electron, muon, hadron, π^0, ...

- Track, shower, vertex identification

Physics we want to do:

- Anything neutrinos can do
- Near to far detector comparisons (for systematics)
- Beam induced exotics (e.g. dark matter)
- Beam induced detector external events (rock or sandy muons)
- Measure cross sections in near detector
- Use data from separate modules/TPCs
- Treat Dune giant modules as quasi-independent, but look over all for supernova – correlate triggers across modules

Calibration & Systematics tasks that are beam related:

- Through going muons
- Timing in trigger with accelerator
- Measure cross sections in near detector
- Measure near detector flux
- Measure dirt backgrounds
- Measure acceptance for side-bands
- Measure incident particle spectrum
- Cross TPC and cross module calibrations
- Near to far relative calibration (but not clear how to do)
- Horn off tests

The simulation and reconstruction software shall record the provenance – specifically the values of parameters used in the simulation and processing.

The software framework shall provide the functionality to run modules multiple times per event with different values of the parameters for systematic uncertainty studies.

Associations shall be possible to follow without knowing the module names that produced the data products pointed to by the associations.

A hierarchy of reconstructed particles – using the GEANT particle tree as an example – shall be represented in the event and traversable by analyzers.

LArSoft shall provide the functionality to measure the energies of beam neutrinos in interactions in the detector.

Interactions produce particles that are observed as tracks or showers.

The analyst shall have the ability to measure the properties of a track, specifically the momentum, the particle ID, and the direction, from the properties of the 3D track or shower.

For a track, the momentum is determined by range, for stopping particles, or by multiple Coulomb scattering for particles exiting the detector.

For showers, the momentum is determined by summing the total energy.

The direction is determined by fitting hits in the detector to a trajectory.

The direction for a track shall be found along the extent of the track. The direction for a shower is found at the beginning, before the shower has developed and looks more track-like.

The reconstruction software shall be resilient against dead and noisy wires.

All reconstructed quantities shall have associated uncertainties.

Analyzers require knowledge of the extent of the track or shower.

Ability to analyze and correlate/tag from TPC and auxiliary detectors.

Merge data from separate TPCs/modules for same spill.

Seamless intermodular reconstruction.

Knowledge of beam conditions (e.g. horn off)

4.2 Hit finding

- LArSoft shall provide wire signal processing to remove noise and to prepare wire signals for hit finding.
- Need to locate a charge deposition on a wire producing a hit object that is characterized by a time (TDC ticks), width, amplitude, and area (ADC * Ticks) that is proportional to the deposited charge. This is accomplished now by fitting analytic functions to the ADC measurements vs. tick. The area shall be provided both as the integral of the analytic function and as the sum of pedestal-subtracted, filtered ADC's. A means of quantifying the goodness of fit shall be provided.
- In order to calculate the area properly, the baseline ADC reading shall be subtracted or otherwise accounted for.
- A calibration mechanism shall be provided for converting from area to deposited charge.
- Need to separate nearby charge depositions into distinct hits. Nearby hits shall be grouped into multiplets that can be merged or preserved in later stages of reconstruction.
- Monte-Carlo truth hits shall be provided, with associations to the Monte Carlo true particle information. In cases where charge from more than one Monte Carlo particle arrives together with another, then the Monte Carlo truth hits shall be reported as separate recob::hits for each particle, even if they would have been combined by a real hit-finder.

4.3 Find Clusters

Clusters may be used as an interim pattern-recognition step before 3D track and shower reconstruction.

Two-dimensional clusters shall be found by grouping hits in a single wire plane. Hits may be shared by multiple clusters. Shower-like clusters and line-like clusters may be handled differently by downstream algorithms. There should be a mechanism for distinguishing a shower-like cluster from a line-like cluster. The properties of a cluster include a beginning and an end in (wire,time) space, and an angle or slope for each end. There shall be a mechanism for associating hits with clusters.

A Monte-Carlo truth cluster finder shall be provided that groups Monte Carlo truth hits from individual particles. We also are interested in forming clusters out of reconstructed hits (which may contain charge from multiple particles but usually just one) that are as pure and complete as can be obtained by examining the MC information in the simulated reconstructed hits.

4.4 Locate Tracks

- A 3D track shall have a beginning and an end in (x,y,z) coordinates, and a direction at both ends. The trajectory between the ends shall be stored with the track data product.
- Associations shall be provided to hits and/or clusters or other data products.
- A momentum shall be associated with a track, and kinetic energy. (? To discuss which of these is more relevant or convenient to use, keeping in mind that the particle ID affects its interpretation)
- The track objects shall provide the necessary discriminant variables to measure energy, direction, and particle ID (see below).

The coordinate systems shall be clearly defined and documented.

- Tracking algorithms must achieve excellent efficiency in order to ... (explore the list the bigger goals here)
- Algorithm developers must achieve an average track-finding efficiency of greater than 95%.
- Algorithm developers must achieve short-stub finding efficiency of greater than 90%.
- need to reconstruct short proton tracks from a few wires
- requirements on short tracks and tracks in crowded area to achieve VTX-01
  - multi-pass fitting with different hit selections and ranking
- noise study
  - ability to run filtering at different levels and run track reconstruction on each of the different configurations, with result ranking
  - modeling of the detector noise needs to be split in independent modules: physics, electronics
  - interface to superimpose measured noise

4.5 Shower Reconstruction

A shower is like a track with a less distinct end, and the trajectory along the path is not well defined or useful. Showers are three-dimensional objects.

- Showers shall have a start point, a direction at its start, an extent, the total energy, the width
- The data product for showers shall be reviewed by shower experts and stakeholders.

4.6 PFParticles

PFParticles are reconstructed Particle Flow Particle candidates.

Hits that are not assigned to particles (tracks or showers) shall be identified as such and accessible to analyzers.

4.7 Vertex finding

LArSoft shall define a vertex object that is the 3D intersection point of tracks and showers in the event. A vertex has a (x,y,z) location and associated uncertainties. It also has associations to tracks, showers, and clusters that contribute to it. A method for associating the end of a track or a shower to a vertex shall be provided. A representation of which end of a track or shower is associated with a shower is to be provided.

VTX-01: Algorithm developers must achieve a primary vertex resolution in three dimensions adequate to meet the physics requirements of fiduciality and PID.

4.8 Particle identification

Algorithm developer must achieve e/γ separation of 90% efficiency for electrons.

Algorithm developers must achieve 99% rejection of photons from π^0 decays using both dE/dx and topology.

4.9 Distinguish and Identify Muons

Roles: Scientific Data Analysts.

The scientific efficiency and precision goals of Muon identification as a result of beam - as well as non-beam events differ for each experiment. These are Science requirements.

Other system functional requirements are common across multiple experiments:

MID-01: Direction and timing information from external detector reconstructed data product shall be available within a detector event.

MID-02: Algorithms must associate external detector information with optical hits, and indicate if TPC hits are associated with an external event.

4.10 Non-functional requirements

- Performance requirements
- Interface and protocol requirements
- Operational and verification requirements

4.11 Verification

Test/Measurement:

4.12 Others

- (from LArIAT) results need to be electronically tabulated to be used by other experiments

- (from LArIAT) need to be able to re-analyze (LArIAT) data
- (ICARUS) need to have information allowing to synchronize the subdetectors, as downstream as possible
- about visual reconstruction tools
  - (ICARUS) visual tool should provide control and manipulation of objects from the lowest level (example: TPC waveform)
  - (ICARUS) visual tool should allow to save and resume a working session
  - the framework should provide run-time configuration
- in iterative processing, need the option to save the results of each iteration, or not
- ability to do a local snapshot of a subset of databases
- energy calibration
  - reconstruct stopping muons with high efficiency
  - need to determine a high-purity π^0 sample
- shared database infrastructure
- provide common databases for physical and mathematical constants
- simulation results should be reproducible and repeatable (stress on random decisions)
- data products should be consistent between the experiments
- ability to correlate generated information with reconstructed objects

4.13 Beam simulation

If you want a percent level measurement across experiments, then beam simulation must be integrated (in terms of organization and responsibilities). Is LArTPC the guardian of this or is this an experiment function? Integration is necessary software-wise – spectrum is common interfaces on one end, other end is full end-to-end beam to detector simulation within the framework.

4.14 Reducing flux systematic

Full simulation of beam line elements and monitors. Do you run G4Beamline inside of framework? The framework can act as a glue with common interfaces to run these components together.

Interface with beam simulations.

4.15 Analyses

Integrate analysis frameworks into reconstruction framework – what is the ecosystem.

Use cases for an analysis framework:

- Get particle lists (e.g. from Pandora)
- Class of algorithms that do full neutrino event reconstruction (typically an analysis, not production)

Why can't you do these things in the reconstruction framework? Reconstruction framework needs to keep ease of installation, build, run, high priority. That typically doesn't happen and people fall out of the reco framework. Hard to get them back even if the reco framework improves.

Need the fast iteration for development.

Requirement: The framework needs to support fast algorithm and analysis development iteration.

Requirement: Have the framework write out appropriate set of objects/data that can be read by analysis tool (e.g. Root) for fast analysis/development.

Requirement: Don't screw up the framework so that people hate it.

Requirement: Full development/reconstruction/analysis chain needs to be considered when designing the framework, etc.

Requirement: Framework should be installable/workable on your laptop

Requirement: Perhaps need an analysis subset of the ecosystem - for working on the plane

5 Overarching Analysis strategies, light detection systems and external detectors

5.1 Description

Overarching Analysis strategies, light detection systems and external detectors includes:

- strategies, (largescale to individual events).
- other stuff, (cosmic ray taggers etc.).
- Data and Dataset management
- Meta-data management
- Analysis techniques
- Analysis toolkits
- Analysis workflows
- Real-time/Data Acquisition

5.2 Monte Carlo interoperability

Requirement: Ability to transfer MC Truth initial 4-vectors for simulated events among different stakeholders of the software. Also for the same rnd seed value, so as to get the exact same track and shower development. (Would that rnd seed be sufficient?) The use case is to have the same event (right down to the track and shower and neutron paths) simulated in LAr which happens to be contained in a TPC with different E-field, wire geometry, light collection, or size and shape.

- Passing of seeds required, or a way to initialize them directly, job-by-job and event-by-event.
- Imposes requirements on the generators and their interfaces.

- For any generator: independently write the outcome of the neutrino interaction, separately from all the downstream interactions and decays of the products (a common format is an obvious knock-on requirement).
- Determine from generators whether the steps of the downstream of the initial interaction can also be output and made available for simulation of different LArTPCs

5.3 Non-TPC, Non-Light-Collection Detectors

Requirement: To make adding new auxiliary detectors smoother, end-to-end. This means a digit-producer which can land its output in a class ready (Must they all be or maybe inherit from the AuxDetDigit class?) for consumption by RECO modules. This could be an extension of the artmod tool. Would the opportunity to deifine AuxDetGeo objects be suggested? Can the CMakeLists be checked to ascertain whether the new code will be compiled? And could a skeleton fcl file be produced so that the developer can isolate their new producer's run-time behavior at its nascent stages?

Alternate to the above: The lardata repository already defines the data types. Let ART handle more than just LArSoft, allowing dedicated software written for the non-TPC detectors to be run alongside and in the same event as LArSoft, knowing perhaps about the LArSoft datatypes (important at the reconstruction stage.) Arriving at a common geometry object for all these presents a hurdle and it must be overcome.

5.4 Development tools ecosystem

Requirement: (mrb) Improved version conflict messages at compile time. Current messages list one extant/available version and one which is being required, but it is perfectly opaque to the user which is which. When the developer resorts to mrb uc and mrb uv commands, and still cannot achieve compilation, how should they proceed?

- Related: (ups) Add tab completion to the ups command. The ups tool interface is pretty rough. The most-used command is "ups list -aK+". The ups command itself only issues the help list, and of course none of the commands it lists are tab-completable, effectively hiding them to most new users.
- Related: (samweb) Exact same violation of the unix tab-complete conventions as above.

Requirement: The ability to distribute noise libraries for jobs which add those to MC, for instance. These are large libraries of data-like files, but they are not treated identically to data files, since they are input to the art source creating noise-overlaid MC.

Requirement (desired): Consolidation of ecosystem tools. Maybe just lar –help, with pointers to the most useful tools (artmod, ups, mrb, dumpers of fcl contents and event files, samweb, etc).

5.5 Heterogenous work-flow environment or framework

Brett Viren's email: (Use Case) Exchange methods are needed to deliver LArTPC DAQ and simulated data to a distributed Wire Cell process and to deliver back Wire Cell results for further processing. The exchange methods need to be implemented in ways that are compatible with the computing architectures that Wire Cell targets.

The working concept for now is influenced by ATLAS's Event Service. Some ”normal” Art/ROOT-full process will run on a ”regular Linux” node at the edge of the HPC or as a special allocation at a grid site running the wider Wire Cell process. This special job will be responsible for receiving ”normal” data, eg Art/LArSoft data products/files and converting it and feeding it, DFP-style, to Wire Cell via a non-ROOT transfer method over the LAN. Maybe this transfer is based on ZeroMQ, MPI or even WAMP - t.b.d. This same process is there to receive results From Wire Cell and convert them back into whatever form (eg, ROOT-based) is needed for down-stream processing. Such a process may need to be tailored for the architecture (eg, what works on Grid may not work on HPC) so some careful thought on interface design is still needed.

Requirement: Actually, how does this generalize to a requirement on the ecosystem?

- Approach one: Easily allow import and export of data as a part of processes (implies queues)

- Approach two: Extend the framework to allow running over a heterogenous hardware collection (some processing is GPU-run, other parts are CPU-run.)

5.6 Detector Data Collection

The detector data collection (DDC) system includes the Data Acquisition System (DAQ) and any information on calibration, timings, etc that is gathered in real-time when the detector is active.

Roles: Algorithm Developers and Algorithm Validators generate these requirements.

DDC-01: A DAQ Event shall represent a data collection period.
The DAQ should package and output to downstream systems information for a full ”data collection period” as defined by the experiment

DDC-02: The DAQ Event must be capable of being split to represent information from well defined smaller time units.
The reconstruction and analysis algorithms may need to be run on data that is a sub-set of an event. New identification numbers (event numbers) need to be assigned to each of the split pieces.

DDC-03: A DAQ event shall contain raw collected data from the TPC, optical readout systems when applicable, and have the capability to include Auxillary detector data.

DDC-04: A DAQ Event shall include related beam and non-beam information for beam experiments.
Including:

- metadata necessary to access beam-related timing information and other beam conditions.
- means to include only optical read-out (i.e.no TPC data).
- means to include only TPC data (no optical read-out).

DDC-05: For beam experiments, a DAQ event shall be capable of directly storing beam conditions and timing information.

DDC-06: Reconstruction data products and other summary data from external detectors event must be *mergeable* with DAQ events to form reconstructable DAQ events.

DDC-07: Prior to any TPC reconstruction activities, a detector event shall include data from all external detectors associated with (collected during) this time period.

5.7 Calibration

Calibration information is needed across a variety of time-scales:

- Have the ability to access and determine the status (for each sub-detector component on a channel-by-channel basis
- initial offline reconstruction and analysis processing in order to apply the right corrections and calibrations to the event data;
- short and medium term to enable tracking of time-dependent changes in the detector performance and characteristics; and
- for the time-scales of future experiments to be able to compare results and detectors.

Roles: Algorithm Validators and Scientific Data Analysts depend on these requirements.

CAL-01: The DAQ Event time information shall be used to identify relevant calibration data.

5.7.1 Laser Calibration

Laser Calibrations (LAS) give detector information achieved through knowing the geometric and energy characteristis of a laser beam sent through the detector systems.

Roles: Developers (Algorithm and Infrastructure), Algorithm Validators and Scientific Data Analysts and Production Operations depend on these requirements.

note: are these requirements laser calibration specific or are they general calibration requirements which Laser Calibrations must also satisfy?

LAS-01: Must be able to calculate calibration datasets based on laser calibration runs.

LAS-02: Must be able to supply algorithms for calculating calibrations based on laser run data. (what form does this input data need to be in?)

LAS-03: Must be able to register calibration dataset by time they are valid

LAS-04: Calibration datasets must be versioned

LAS-05: Calibration datasets must be accessible from reconstruction applications running on the grid. (from where?) Running on laptop. Directly from the data files being processed.

LAS-06: Calibration metadata must be stored in each data file that uses it. (this is actually not such a good requirement - it assumes files are necessary)

5.8 Purity analysis

Data products and procedures that permit purity analysis shall be available. (SDA) (this is not stated right at all and requires much discussion - what are these things?)

5.9 Design studies

5.9.1 Dual phase TPCs

Simulation shall provide the capability of studying dual-phase TPCs, as well as current single phase designs. (SDA) What does it mean to do this? What needs to be available?

5.9.2 readout configuration

Simulation shall provide runtime configuration parameters for what? (SDA) Wire pitch and angle? Things like this or not? position of TPC readout plane?

5.10 Analysis Strategy

Goal: Event reconstruction based on topological and calorimetric reconstruction on the event information

- The ability to access associations between various reconstruction algorithms (Pi0 identification, photon) and stitch together

Use case: (Pi0 identification) - Take in shower and track (whether from MC or data) objects - Run an algorithm which identifies photon showers - Run an algorithm which associates photon shower pairs from a common vertex - Run an algorithm which makes “Pi0” object from associated pairs of photons - Create “Pi0” object which has data members such as energy, momentum, mass, location (x,y,z).

Use case 2: (Pi0 identification w/ different photon algorithm) (all algorithms have standard output so trying a different photon algorithm is straightforward)

- LArSoft shall know how to do associations between objects without the user doing all the archaeology

Example: If I have the Track Handle, then LArSoft should know how to handle what clusters and hits are associated with the Track Handle.

- Definition for every data product explicitly written somewhere

Goal: Having build times less than 30 seconds

Use case: User wants to look at data with minimal processing and overhead directly from the production file.

Requirement: The standard file format must be able to be used independently from the processing framework.

Use case: Experiments in the SBN program will do joint analyses.

Requirement: Joint experimental analyses through the use of data products must be possible.

5.11 Monitoring

- real or near -time for validation of the event data and monitoring of the detector performance;

5.12 Non-functional requirements

- Performance requirements
- Interface and protocol requirements
- Operational and verification requirements

5.13 Verification

Demonstration and Test/Measurement: Histograms and plots based on standard analysis and reconstruction steps and regression tests in downstream systems.

5.14 Dataset Management

Use case: Read data taken years after the fact and rerun the software from that era. The data could potentially be from previous experiments that are no longer running. This puts some requirement on the experiments to preserve databases, etc that are external to the code.

Requirement: Must retain ability to read data over long periods of time. There should be a mechanism to translate older data formats into modern data formats.

Use case: Need ability to access experimental dataset from distributed sites. Example distributed sites are the grid or cloud for processing.

Requirement: The distribution of the data must not become a processing bottleneck.

Use case: Performing calibrations may require the ability to look at triggers in sequential order.

Use case: Analyses may desire non-sequential access of the data. Potential ideas were inverse beta decay, muon decay, delayed neutron interactions. Most of the processes that we thought about were short compared to the readout time of a LArTPC.

Requirement: If time-ordered access of the data is required, that access need not be efficient.

5.15 Metadata Management

Use case: Comparison of multiple analyses. Important information to capture are: the software versions of the code used, including software external to the framework; the state of the conditions system such as calibrations, alignment, and hardware settings; data quality information to flag times where the detector performance was acceptable for analysis.

Requirement: The data should link to information about how the data were produced.

Use case: smaller experiments do not have the available effort to design individual solutions. The SBN program will want to access common condition storage. Use of conditions data in common reconstruction algorithms demands a common interface.

Requirement: A common interface for specifying and accessing conditions data is necessary.

Use case: ability to access conditions data without an internet connection.

Use case: find data satisfying specific conditions, including supernova triggers, proton decay, or beam conditions such as intensity and profile. Requirement: the framework must supply the ability to tag data satisfying specific conditions.

6 Human interactions, computing systems, software and interfaces

Use cases:

- Support workflows from experiments
- Collect workflows from experiments
- Detector design
- DAQ/Trigger design
- Reconstruction
- Analysis
- Online/offline monitoring
- Perform Calibrations
- Support 3D maps:
  - Electric field
  - Photon map
  - Space charge

Requirements:

- Support the appropriate parts of workflows for the offline software.
- Support R&D activity aimed at exploiting new technologies such as
  - GPU, MiCs, and FPGAs
  - Deep Learning
- Long term: Support all relevant forms of ”Concurrency”,
  - Processing multiple ”Events” within single algorthim. For example, ability to simultaneously perform the FFT deconvolution for multiple events on a GPU.
- Run and compare reconstruction, potentionally one algorithm at a time, from other experiments (e.g. ICARUS and WA105) within the framework.
- Visualization is tightly coupled to the offline software (i.e. feeds back into offline) and keeps providence of the human interactions. Examples: hand scans (similar to ICARUS and ArgoNeuT), debug/develop/characterize reconstruction, train Machine Learning algorithms.

- Visualization system uses a web browser.
- Migrate to parallelized art.
- Fully reproduce the DAQ/online filtering/triggering for designing, characterizing, and simulating the full data flow.
- Understand the required performance (CPU, Memory, ...)
    - Measure and record
    - Project to future scales (experiment input?)
    - Efficiently store 3D maps, including interpolation, compression, ...
- Use standard configuration language, preferable python.
- Python algorithms
- Single configuration mechanism. For example interfaces to external software includes a configuration.
- Automated data quality monitoring.
- Well defined and agreed for transitioning ownership of components.
- Well defined and agreed mechanism for making decisions.
- Effort accounting aimed at appropriately assigning credit and keeping track of contributions.

6.1 Description

Human interactions, computing systems, software and interfaces includes:

- Visualization
- Scientific and development workflows - including human components
- Regression and validation software/processes
- New computer hardware architectures and multi-threading
- Software frameworks and interfaces
- Organization of common/shared components, including policies
- Shared software documentation

6.1.1 HPC + parallel'ization use case

- Huge data volumes, so storage issues
- Reprocessing times desired to be reduced
- OSG
- Appreciation for the fact that code should be written with parallelization in mind. And most of ours has not been.
- External supercomputing resources remove flexibility of OS, compilers, ...

6.2 Visualization

6.2.1 Use Cases

- Online Event Display in control room. *Roles:* Shifters, DAQ experts *Functionality:* Connects to Online data stream, examine for functionality, zooming into desired scale, automatically updating. Computationally efficient so as to draw a new event: less than few seconds to render, minute to reload new event.

- Offline Detector Diagnostics *Functionality:* Ability to select historical events with, say, a file browser. Find interesting events via Run/Event, Time period, Data Quality, Trigger types
- Tuning the reconstruction *Roles:* Analyzers, Developers. Draw recon'd objects with/with-out raw waveforms overlaid. Re-recon the event again with new inputs or reconfig. 3D display. Require the 2D objects to obviously correlate (via color for example) to the 3D display. Needs to walk through events without falling over!
- Interactive EVD for Handscanning Studies Separately, we want a potential interactive EVD step. We want to be able to pick features and form new objects. We want to be able to output objects onto the event from the EVD. (Lots of discussion whether this is tenable for large data samples.)
- Public relations 3D display required. Rotation, re-orientation required. Ability to create movies. VR compatible. Accessibility to files for event viewing from the public domain: URLs. Mobile phone app (not necessarily required!)

6.2.2 Debugging and tuning

Roles: Algorithm Developers and Algorithm Testers generate these requirements.

DBT-01: Interface with 3D interactive visualization shall be provided.

DBT-02: Visualization shall display reconstructed tracks, showers, and associated properties

DBT-03: Visualization shall permit reassignment of hits to recognized patterns.

DBT-04: Visualization shall be capable of displaying optical information, as well as information from the TPC.

6.2.3 Part of Reconstruction Workflow

Roles: Scientific Data Analysts depend on these.

6.3 Event-picker

Roles: All users (roles) depend on these capabilities.

EPK-01: A means to register individual events ”of interest” must be provided.
This registration should provide the ability to drill down and access all information and provenance about the event - its history, processing results, associated (versions of) calibrations etc.

Physics groups can input identification information as interesting. For example, a 2nd pass of a reconstruction may wish to mark particular events as interesting. We're imagining ”Event metadata.”

EPK-02: Searchable index of events based on a variety and extensible set of criteria Yes. If we do above, we want this.

6.4 Preservation of Test Beam Experiment Data

The use of large-scale Liquid Argon TPCs for neutrino detection results is entering a new regime. Later experiments and tests will depend on access to the data, calibrations and analysis of previous data collections in order to understand and validate the details of new information (e.g. changes of the detector and medium characteristics over short to long times; effect of dead areas/wires to the accuracy and efficiency of the results).

Thus preservation of all this information is a functional requirement

Roles: Needed for Scientific Data Analysts.

PRE-01: A preserved repository of LAr performance characteristics accessible through programmatic and human interfaces.

PRE-02: The repository will include previous test beam experiment information and be maintained ”for the forseeable future”.

PRE-03 The repository data and administration must be accessible through a web interface.

PRE-04: The repository shall provide sufficient information to know how to retrieve raw data that produced the measurements.

PRE-05: The repository shall provide sufficient information to know how to retrieve the code versions that produced the measurements.

PRE-06: Curation processes shall support the upgrade data formats to current versions.
PRE-07: The repository must have programmatic and human interfaces for uploading new versions of any data/information.

6.5 Documentation of shared components

6.5.1 Use Cases

- The algorithm user needs to reference Algorithm author’s documentation. Physics note, not code details of interfaces. Require User’s Guide style of information.
- Some external-to-the-ecosystem note which becomes a reference. Such as a FNAL tech memo or a MicroBooNE Public Note or arXiv or Inspire.
-
- Notes with code details: units, function signatures, interfaces,

6.5.2

- content detail requirements
- accessibility and visibility requirements
- quality assurance and enforcing protocol and requirements

6.6 Non-functional requirements

- Performance requirements

- Interface and protocol requirements
- Operational and verification requirements

6.7 Verification

Demonstration: The verification of these requirements includes time-and-motion studies of individual users interacting with the visualization system.

Test/Measurement: Continuous sampling of the information, application of regression and analysis suites, and comparison of output plots,histograms and physics quantities.

A Glossary

Functional requirements: are the fundamental or essential subject matter of the product. They describe what the product has to do or what processing actions it is to take.

Nonfunctional requirements: are the properties that the functions must have, such as performance and usability. Do not be deterred by the unfortunate type name (we use it because it is the most common way of referring to these types of requirements) — these requirements are as important as the functional requirements for the products success.

Design constraints: impose restrictions on how the product must be designed. For example, it might have to be implemented in the handheld device being given to major customers, or it might have to use the existing servers and desktop computers, or any other hardware, software, or business practice.

User requirements: define the results the users expect from the system: “The homeowner shall hear an alarm when smoke is detected.”

System requirements: define what the system must do to satisfy the users: “The alarm will produce a sound between 125 155 dBA”

Design requirements: define all of the components necessary to achieve the system requirements: “The alarm will be produced by part # 123-45-678.”

B Use Cases - working set

Use cases, or scenario, are useful for identifying requirements. They provide a good backdrop for focusing requirements gathering discussion and they also help set the scale or level of detail that is desired. Below are the set of use cases that have been already collected from participating experiments. These use cases only represent a sample of work that will accomplished at the experiments.

B.1 LArIAT

B.1.1 Use Case: Time stamp assurance

Goal: Ensure timestamps across many different systems, digitizers can be gathered and aligned to determine which particles from the external and LArTPC detectors are associated (belong together).

Scenario:

1. Read out all data from LArTPC in 4 sec gate window and identify particles.
   - a. Reconstruct tracks: energy, direction, ID as well as possible.
2. Gather all data from external detectors where resolution on the time is ? $\approx 5\,\mathrm{ns}$ (cosmic ray tagger, neutron detector)
   - a. Tag the ID of the particle and its properties (momentum, position etc) Assume 100% efficiency although in practice almost but not quite 100% efficient.
3. Sift through all this data for a best timing alignment match of the particles (fast readout (e.g. PMT scintillator paddle output) to slow readout components (e.g. LArTPC))
4. Compare the ID of the (external detector) tagged particles to the particle detected in the LArtpc to determine efficiency of LArTPC detector (and s/w)
5. Optimize and iterate 3-4 to achieve best match, thus aligning the timings the best

Validation/Tests:

Inspection - e.g. if the algorithm has selected events with only 1 track scan to see how many a person would have identified.

B.1.2 Requirements on software/computing system

- System needs to handle continuous readout and stream of data/information where the event boundaries are defined within the system not externally.

B.1.3 How results are propagated to the other experiments

B.1.4 Additional Information

- Spill is the only natural event contains several interactions.
- Published results only rely on the s/w no human in the scientific work flow.
- Simulation G4 fine beam line timing structure is not available; so only 1 track is every provided by the simulation.
- Currently available simulations are not sufficient to give accuracy of information needed to use these for validation in Lariat or other experiments. E.g. interactions with lighter nuclei are not accurate when extrapolated to Ar.
- Another run in Feb for a few months. Small R&D questions to answer may be future short runs.
- Light simulation/photon libraries: need simulation of how many times the photons bounce on the reflective foils before landing to the photodetector; not precisely known how detailed the time structure of the PMT pulse needs to be.

B.2 MicroBooNE

The overarching scenarios that MicroBooNE implements are:

1. Reconstruct track-like particles ($\mu/\pi^{\pm}/K^{\pm}/p$).

   - (a) Trajectory.
   - (b) Calorimetry (dE/dx).
   - (c) Particle identification.

2. Reconstruct (electromagnetic) showering particles.
   - (a) Shower vertex and direction.
   - (b) Shower energy.
   - (c) Electron/photon separation using initial dE/dx.
3. Flash reconstruction (optical reconstruction).
   - (a) Time-zero determination and beam gate coincidence.
   - (b) Flash-track and flash-shower matching.
4. Cosmic ray identification/rejection.
   - (a) Geometric identification of cosmic ray.
   - (b) Optical identification of cosmic ray.
5. BNB neutrino interaction identification/reconstruction. Average BNB neutrino energy 700 MeV.
   - (a) CC muon (anti-)neutrino.
   - (b) CC electron (anti-)neutrino.
   - (c) NC.
6. Supernova neutrino identification/reconstruction from supernova data stream. Supernova neutrino energy 10 MeV.

Goal: measure the low energy electron excess. Compare with expectation from non-oscillated neutrino beam. Electron-neutrino appearance.

B.2.1 Use case: Reject cosmic rays all in time and out of time

- 10 cosmic rays per TPC readout. $\times 5$ expected number of neutrino interactions
- Trigger rate dominated by cosmic rays
- Integration time is $2.2\,\mathrm{ms}$ with lower high voltage. $1.6\,\mu\mathrm{m}$ integration time compared to the BNB
- Triggered events in time with the beam gate 20% of triggered events will have a neutrino interaction.

Scenario:

1. Full TPC reconstruction optimized for cosmic ray 3-d reconstruction reconstructing track like particles; high efficiency detection of long tracks.
   - Short tracks less efficient and/or less precise
   - Showers are less efficient and/or less precise (energy, location)
2. Reject in time cosmic ray as well as out of time cosmic rays.
   - Out of time can be rejected by matching to an out of time PMT pulse.
   - In time only handle is the reconstructed feature of the track. Cosmic rays will enter and leave the TPC and will in general be more vertical than the neutrino interactions.

B.2.2 Use case: Identify Neutrino interactions

Scenario:

1. Reconstruct particles, different optimization than for cosmics, particles that originate in volume of the TPC which may or may not exit.
2. Reconstruct vertices from which multiple tracks (including muons, protons) are coming
3. Understand showers which come as a result of a vertex.
4. Identify pi-zero and photon induced showers displaced from the primary vertex. Identify gap between the shower (pi-zero there are 2 showers) start and a second interaction vertex.
5. Calculate de/dx of the shower start (2cm) to discriminate between electron and the photon (electron and positron both contribute to the shower) induced showers.
   - Resolution for electron showers at low energy.: Energy and direction/range of the Neutrino itself. Down to few hundred MeV neutrinos.

Reconstruction Steps:

1. Optical reconstruction. Determination if an in-time particle at all (at start when no trigger)
2. Filter on result
3. TPC Waveform filtering filter out the noise and shape (fourier domain) the pulses to be positive going.
4. Hit finding identify pulses by hit time and area (total charge in the hit)
5. 2-D reconstruction clustering finds hits close together into a collection
6. A or B Pattern Recognition only. with what classification error profiles none that we know of. If the shower or track refer to the same region of space this is allowed tracks are discovered inside of showers eventually will discard such tracks. But showers can be delta-rays from muons so not clean/clear
   - a. Pandora reconstruction starts from hits and make 2-d and 3-d reconstruction. Cosmic ray optimized and neutrino interaction optimized pass. Distinguishes tracks and showers in the output)
   - b. Independent pattern recognition Cluster 3-D, (may only be optimized for tracks, not clusters).
7. Independent shower and track reconstruction 3-D with error matrices with goodness of fit etc.
   - a. Calorimetry measure energy deposition de/dx to distinguish muons, protons, pions
8. Energy estimation of the shower and track like objects.
   - a. Independent shower energy reconstruction
   - b. Contained tracks normally have energy estimated by the range.
   - c. Non-contained tracks are much harder to estimate the energy can estimate through multiple-scattering measured. Theoretical description of how particles scatter proportional to momentum. With long enough track can use this. (1m track lengths for example)
9. Classify/ identify Particles. Anything displaced from the vertex is used to improve the energy reconstructed for the neutrino interaction.

- a. Muon low ionizing relatively long track. May escape
- b. Hadrons/protons higher ionizing/short tracks
- c. Pions difficult to distinguish from muons.
- d. Electrons showers coming from the interaction vertex, prompt showers.
- e. Photons displaced showers (background for electrons if the displacement is very small). Use de/dx discrimination to reduce this
- f. Neutrons/neutral Ks/ lambdas very hard to identify. Can have displaced vertices. Neutrons from the nucleus are not rare cant identify them though. Not reconstructed per se.

Testing/Validation:

- How does one determine the accuracy and/or efficiency? All have mis-identificatin probabilities. Combination of simulation, control samples e.g. LArIAT results with control beam.
- Any hand-scanning comparing human identified hit/track with the automated reconstruction; Automated reconstruction to perform as well as hand scanning. Both case by case and statistically.
- Data streams with rare events e.g pi zeros as calibration sources.

B.2.3 Use case: Determine rare backgrounds

1. Cosmogenic events, dirt events having something interacting round the detector or in the wall with generated particles entering the TPC.
2. Need to estimate the backgrounds independently.
3. Material internal decays Ar very low energy, below threshold. Few MeV.

Reconstruct tracks down to the shortest possible length e.g. few -10 wires. 10-20 MeV protons. No threshold there.

Categorization parameter for cosmic/non-cosmic tracks. Cosmic score to every track like particle. Used for removi

B.3 ICARUS

B.3.1 Optical detector reconstruction

- many more PMT than in the past (≈ 90 per side)
- fundamental to extract regions of interest
- $50\,\mathrm{ns}$ resolution was not enough (on what?)

B.3.2 Interactive analysis

- allow to select the objects to work with
- depending on the conditions, could go down to raw data
- could “save” at any point

B.4 SBND

B.4.1 Use case: Muon direction

Goal: Have efficiency of recognizing the muon direction to be 90-95%, based on detector, beam and DAQ components.

Scenario:

- Use external detectors detectors (muon tagger, photon detector..) and TPC information, linked via timing signals, to determine whether a muon track is from cosmic ray or due to a beam interaction.
    1. Collect timing signal from each of the hits in the tracking system (5ns time window). There is a common clock signal..
    2. Link TPC start of the drift window and the timing of each individual hits.
    3. Collect information on pulses from all in the same time window/from the same time.
    4. Link timed pulses
- Determine direction of the muon
    1. Calculate where the muon crosses the boundary of the TPC, check consistency with a time tick on the wire.
    2. Muon tagger knows the position of the track.
    3. Compare light signals external and internal to the TPC. The Differences in the signals tells you whether the muon is entering or exiting the detector. Some muons that are ionization tracks look like they enter and stop.

Validation/Tests: Compare to MC.

B.4.2 TBA

- Analyze the beam T0 timing signal in order to get the best resolution.
- Will data from both detectors arrive in the same bundle dont know. DAQ question. Must be combined in a single data product for that event. Whether it is merged after DAQ is not known

B.5 SBN

B.5.1 Overview

Varying and different signal to noise ratios across the electronics and noise and dead channels across the TPCs may lead to different algorithms and/or implementations for particle ID.

B.5.2 Use Case: Fast monitoring of detector and electronics characteristics

Scenario:

The quality of the physics output can be improved by fast (¡1 day, ¡1 hour, turnaround) processing of data using the offline system, possibly with less precise algorithms being run. As examples:

- Electric Field distortions can be measured using cosmic muon events - not event by event, but immediate turn around with less sophisticated reconstruction on a selected sample of data.
- The noise level of the wires as well as looking for dead/broken wires determined through offline processing of a selected number of events near- or on-line.
- Evaluation of the purity of LAr using cosmic muons. Need offline processing runs with less refined algorithms that can run quickly on a selected sample of events and tracks.

B.5.3 Use Case: Comparisons using different MC techniques against the detector data

Scenario:

- Use flexible access and ease of configuration to apply different MC techniques in the offline processing systems. These are needed especially during commissioning of the detector where the many unknowns in its performance, characteristics and detailed efficiencies (e.g. Dead zones).
- Deeply investigate every feature of the detector and make modifications to the offline systems to accommodate and mitigate them.

B.5.4 Use Case: Analysis and characterization of detector, electronics and read-out conditions

Scenario:

- When encounter unexpected issues with the data and detector need to be able to check a sub-set of events by eye, interact with individual components and parts of the data, and make rapid deep investigations of the details of any/all of the information available. Similar interactivity is needed for complete validation of the events in the final datasets.
- Perform multiple MC runs with differing parameters to investigate different possible error conditions in actual data. Train the monte carlo generation and analysis algorithms across a wide range of different detector conditions.
- Run modified algorithms for the light detection system to accommodate ranges of time and detector specific space resolutions dependent on the granularity of the light collection systems.

B.6 35t

TBA

B.7 ProtoDUNE

B.7.1 Overview

Experiment is a test and needs will be similar to LArIAT with the differences including that the LArTPC is much larger, the presence of APAs, and the data volumes. The needs include time-matching of the external detectors to the LArTPC readouts.

Physics goals include particle identification, testing reconstruction algorithms to be applied in DUNE and measuring the performance of the detector and identifying any issues in the detector design. The detector is on the surface not underground and thus the cosmic background is a lot larger.

B.7.2 Issues

- Part of the TPC has the electric field in the opposite direction and some space which is field free — there will be no drifting electrons in this space, but photons will still be generated.
- How to define an "event" from a set of data collected during a spill (cf. LArIAT). Time gate is 45 secs cycle time and two 4.5 seconds spills. Sprinkles of particles during the spill that can be adjusted. Beam includes pions, protons, kaons, muon, and some electrons mixed together.
- Testing of Geant4 outputs – also like LArIAT. Need tracking efficiency; not drop a lot of information on the floor.
- 25 terabyte of actual data. 2 months of commissioning data and cosmic rays. So plan for order of magnitude larger: ≈ 0.5 petabyte total of tape at CERN just for raw.
- Since one of ProtoDUNE goal is to test the algorithms there may be multiple algorithms run; will have results from MicroBooNE so perhaps not as many as MicroBooNE.
- Will need to integrating at least 2 geometries (move drift length exploring design features) of ProtoDUNE into the reconstruction
- Will need to compare single and double phase against each other.
- Incorporate the laser calibration system; monitor liquid purity and temperature — need those hooks easily in the reconstruction system.
- Analysis code sharing e.g. purity analysis if you are using tracks to get the purity extracted.
- More sharing of analysis code. Landau Gaussian fits e.g.
- Easier utility for analysis type code. E.g. cut out the corner of a detector because it does not work. Need to communicate this to another experiments.
- Does one need an analysis data model that is different from the standard Larsoft one for reconstruction.
- For 35ton last collection wires in APAs cut out in calculating dE/dx: they collect more charge than the other wires, and their signal may be displaced in time.
- Electrodes in the gaps modify the field shape and affect the electrons. Does not just affect the last wire but also others. May have to simulate edge effects more carefully. Geometry utilizes a simple formula based on wire pitch and wire number to find the wire position, but this approach may not work for the ends of the wires.

- Space charge for ProtoDUNE – unique issue together with LArIAT, but LArIAT is smaller. The particle beam itself directly creates space-charge while crossing the detector, since it's made of charged particles. Space charge modeling needed for ProtoDUNE longer drift and extra pitch and laser calibration. New pattern of analysis. What larsoft needs to do. Field map reconstructed from the laser data. The procedure may be experiment specific. Physics is the same. Simulations the same.

B.7.3 Use cases: Deconvolution of space charge

Goal. Deconvolve the effect of space charge during reconstruction. Reducing the effect is a requirement on the detector, analysing the residual effect is software.

Goal: Rerun reconstruction on ProtoDUNE data during lifetime of DUNE.

Scenario:

1. Preserve the data and the software.
2. Maintain the Calibration information for a decade or more.
3. Test extrapolation from one detector to another

B.8 WA105

B.8.1 Overview

Test beam experiment to characterize the response of large double phase LAr TPCs. Given known particle energy in the beam measure the efficiency of reconstruction.

Beam has well defined primary energy 0.4-12 GeV. Measure and check the energy resolution. How much e.g of a 2 GeV incoming beam one can reconstruct in the TPC. Characterize the LAR as a medium Test response of the double phase design as well as the signal to noise ratio on the surface not underground. Beam arrives right at the entry of the fiducial volume Showers should be contained inside the fiducial volume.

B.8.2 Use Case: Characterize the signal to noise profile across the TPC with little drift to 6 metres drift.

Scenario:

1. Use Cosmics and select events at the bottom of the TPC and find their 3-d location
2. Measure the signal to noise ratios

B.8.3 Use case: Test pi-gamma separation and event reconstruction.

Scenario:

1. Run reconstruction algorithm on electron-neutrino free events. Algorithm should identity only pi0 events.
2. Run reconstruction algorithm on pion events since they look similar in topology to neutrino interactions

B.8.4 Challenges to date

1. Biggest challenge will be from surface operations and separate the beam events from all the cosmic events that will get. Few KHZ of cosmics within the drift window of 4 msec after the beam. Few tens of muons in the fiducial volume. Discriminate beam events from the overlapping cosmics.
2. Data throughput 1 PB a day of data. Online Data processing.

B.9 DUNE

B.9.1 Overview

The reconstruction goals for DUNE include:

1. Vertex position resolution: 2.5 cm in all three dimensions. (probably need better than this in order to have e/γ separation topology performance)
2. Tracking efficiency $> 95\%$
3. Short-sub finding efficiency (10 hits or more, all views together): $> 90\%$
4. e/γ separation: 90% efficiency for electrons, 99% rejection of photons from π^0 decays using both dE/dx and topology

The following assumptions are taken from the DUNE CDR Volume 2, Table 3.3

- muon detection threhold: 30 MeV (KE)
- muon angular resolution: 1 degree

- charged pion detection threshold: 100 MeV (KE)
- charged pion angular resolution: 1 degree
- stopping track energy resolution: 5%
- showering or exiting energy resolution: 30%

- electron detection threshold: 30 MeV
- photon detection threshold: 30 MeV
- EM shower energy resolution: $2\% \oplus 15\%/\sqrt{E}$ where E is in GeV
- EM shower angular resolution: 1 degree (ed. note this may be asking a bit much for low-energy showers)
- EM energy scale uncertainty: $< 5\%$

- proton detection threshold: 50 MeV (KE)
- proton energy resolution: 10% for $p < 400$ MeV, $5\% \oplus 30\%/\sqrt{E}$ for $p > 400$ MeV, where E is in GeV
- proton angular resolution: 5 degrees

- neutron detection threshold: 50 MeV (KE)
- neutron energy resolution: $40\%/\sqrt{E}$ where E is in GeV
- neutron angular resolution: 5 degrees

other particles (K, Λ, Σ, deuteron)

- detection threshold: 50 MeV (KE)
- energy resolution: $5\% \oplus 30\%/\sqrt{E}$
- angular resolution: 5 degrees

B.9.2 Use case: δ_{CP} efficiency plot or Mass Hierachy

Goal End-to-end fit — ν_e, ν_μ — separately for near and far and 2 beams — 2 horn currents, anti-neutrino and neutrino beams. 8 samples go into end-to-end fit. Each sample is number of detected neutrinos vs neutrino energy.

Additional samples that measure neutral current and cosmic rays for background calculations.

Fast MC parameters how much neutral current comes in. Mis-identify pions as electrons. Need to exclude neutral currents as much as possible

Detect events and reject the neutral current backgrounds.

Scenario:

For ν_e part

1. Reconstruct hits, tracks and showers
   a) Some events only have showers
2. Make a multivariate algorithm to tell you whether the showers are electrons or photons or pi-zeros. Find the vertex for some channels (use tracks — 1 or more)
   a) Sometimes proton comes out as well as an electron.
   b) Used for rejection of π^0 also, low energy neutrino scattering
   c) Different algorithms for different topology events
3. Second multivariate algorithm that uses vertex and output from first multivariate algorithm for some events.

For ν_μ part

1. Reconstruct hits, tracks,
2. Mis-identify pions/reject them.
3. Muon energy, range and topology from dE/dx if the muon is contained

The following requirement was stressed LBNC review of DUNE on September 2015:

- 80% efficiency for automated reconstruction for quasi-elastic, resonant elastic scattering and deep-inelastic scattering events

B.9.3 Use case: Proton Decay

B.9.4 Use case: Supernova

Supernova detection is a primary science goal of DUNE.

B.9.5 Use case: Neutrino Atmospherics

B.10 Historical experience

This information is background information on the human interaction component of processing and analysis from previous experiments. It is not part of the requirements gathering per se.

B.10.1 Use case: Reconstructing tracks with increased efficiency and minimizing errors

- Humans play an important/critical role.
- Tracking reconstruction was not good sufficient .
- Semi-automated human scanning good for Pattern Recognition to recognize a track
- Especially used where there are many tracks.

Scenario:

1. Run Crude reconstruction and simple cuts to select the event interested in.
2. People then scanned the events.
   - a. Use tool mouse movement, clicks - to do reconstruction on the event display.
   - b. See Hits on different views that belong to a track. - c. Include the hits using a mouse and click to say that is part of the track. - d. Total 3,000 events and scanned sub-sample.
3. Run more fine grained reconstruction

Validation/Tests:

Aware there is a bias due to human component of the scientific workflow for results.

- Estimation of the systematics/random errors not done completely due to lack of manpower.
- Also scanned MC as well as detector events and used results to evaluate efficiency. This efficiency was input to the final analysis.
- Some data multiple people scanned, compared results and then got together to make them consistent.

Source: Icarus@LNGS Reported in published papers:

- Description of the procedure to preselect, identify electron neutrinos and how systematics was crosschecked on MC: Experimental search for the LSND anomaly with the ICARUS detector in the CNGS neutrino beam, The European Physical Journal C(Eur. Phys. J. C (2013) 73:2345), arXiv: 1209.0122
- 3D algorithm, tests on single tracks of stopping particles selected in CNGS data: Precise 3D track reconstruction algorithm for the ICARUS T600 liquid argon time projection chamber detector. M. Antonello et al. (ICARUS Collaboration). Advances in High Energy Physics, Volume 2013 (2013), Article ID 260820, 16 pages. Also on arXiv:1210.5089v2 [physics.ins-det]

B.10.2 Use case: Manual processing

- Similar to Argoneut.
- Manual corrections mixed with automatic processing mainly for analysis of multiple scattering, especially with muon candidates, tests of PID and recombination on hadron and muon tracks.

- Mostly manual workflow for selection of cascades, collection of fragments, selection of associated vertex (pi0 decay).
- In general MC analysis was completely manual. Detector data reconstruction manual workflows supported mostly pattern recognitions and selecting of the clusters. The same procedure for MC and data.

Scenario:

Tracks:

1. Run noise removal (some stable patterns automatic, then manual selection of remaining noisy regions), hit finding and fitting, basic clustering.
2. Needed to account for ongoing changes in the patterns of noise. Never possible to prepare filters to remove all automatically.
3. Hand correct connections between Clusters.
4. Merge those that are broken by mistake and edit even single hits.
5. Thousands of muons in the detector. Used as part of the track before it stops by cutting out some length at the end of the track. Estimating the muon momentum and compared to full information from charge deposits of the stopping muons.
6. 3-d reconstruction looked at tracks prepared this way.
7. Note: there was a feature in the software to run through all events and produce gif images of hits or clusters or 3D tracks projected to 2D this was very useful to speed up scanning for reconstruction errors.

Physics analysis for sterile neutrino:

1. Electron events manually scanned to find candidates.
2. Calculate de/dx of initial part of the electron cascade. Very difficult to automate. No algorithm to do this very busy and very complicated. Only by looking at them can tell which can be regarded as the initial part of the electron candidate.
3. Conditions to cut background and calculated manually de/dx on each wire of candidate cascade.
4. 3 people.
5. And on MC.

Validation/Tests:

- Sample of full MC processed in the same way (manually) as data.
- Check the procedure rebalanced cuts were applied to large sample of MC truth. No simulation of the detector of reconstruction steps. Just to events in the detector.
- Cross-checked the cuts which results in the event selection.

B.10.3 Input to the Future

- Tools provided single click inclusion, move in or out, for 2-D images. Would be nice to think of this for the 3-d also.

- Difficult part is the vertex region. Can only look at a single TPC at any time. For detectors with multiple TPCs need to scan across all of them in a single display. (35Ton, ICARUS T600 has 4 physical TPCs (one can think of 8 TPCs geometry for the software purposes))
- Recommend (LArSoft) include some common structures to keep the full event description together with the evolution on information about the particles detected. Common rather that each experiment having to develop their own. There are currently weaknesses of representations - currently associations but an event could be fully described.
- Include full granularity of the event and how it all fits together especially with multiple TPCs.

B.11 Cross-Cutting

TBA