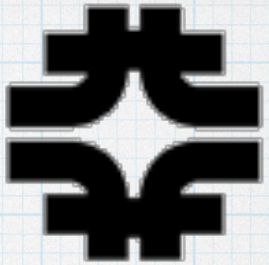


NuTools & LArSim - LArSoft's gateway to GENIE and Geant4

Robert Hatcher
Fermilab Computing Sector
Scientific Computing Division / Systems for Scientific Applications /
Scientific Computing Simulation / Physics & Detector Simulations
2016-06-23

What is usability, anyway?

usability \ˌyü-zə-'bi-lə-tē\ *noun*

"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use."

— ISO 9241-11

effectiveness how fully the final solution satisfies the original need

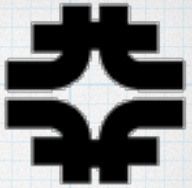
- can you get your idea implemented?
- e.g., signal processing, image processing, MVA...

efficiency how easy it is to get to that solution

- fitness of the tools
- learning curve
- maintainability

satisfaction by how many years working with it will shorten your life

Did I say, “maintainability”!

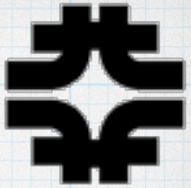


What maintainability has to do with all of this??

... it’s not usability... it’s just code maintainers’ business! *Right?*

Well... no:

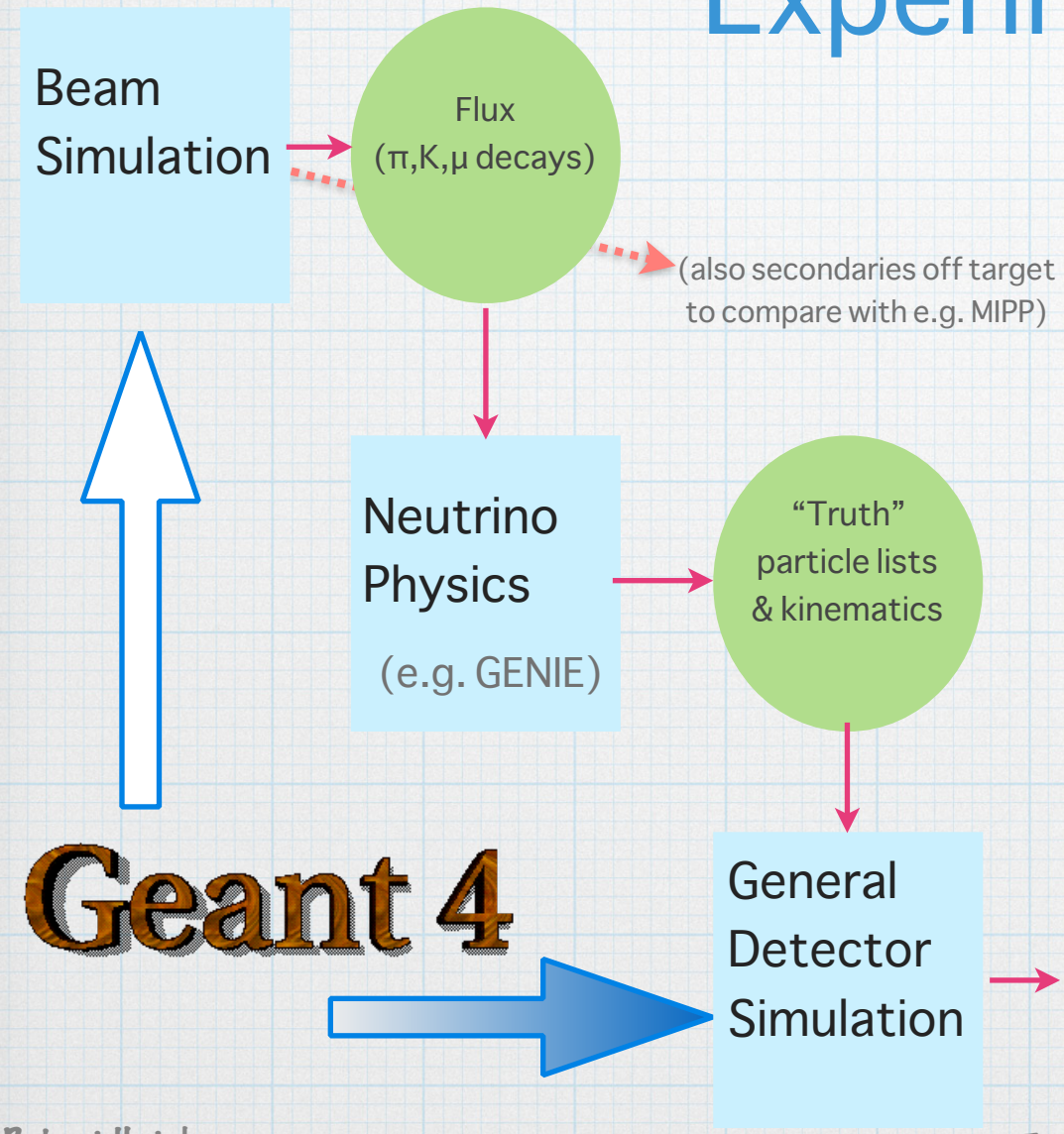
- LArSoft is a collaborative contributed project:
  - *you* write it
  - *you* change, fix and extend code to new needs
  - *you* get frustrated when the code is unreadable
  - $\Rightarrow$ *your* effectiveness, efficiency and satisfaction are on the table
- maintainability is (also) about
  - *design* accommodating changes $\rightarrow$ *effectiveness*
  - *readable* and understandable code $\rightarrow$ *efficiency*



Outline

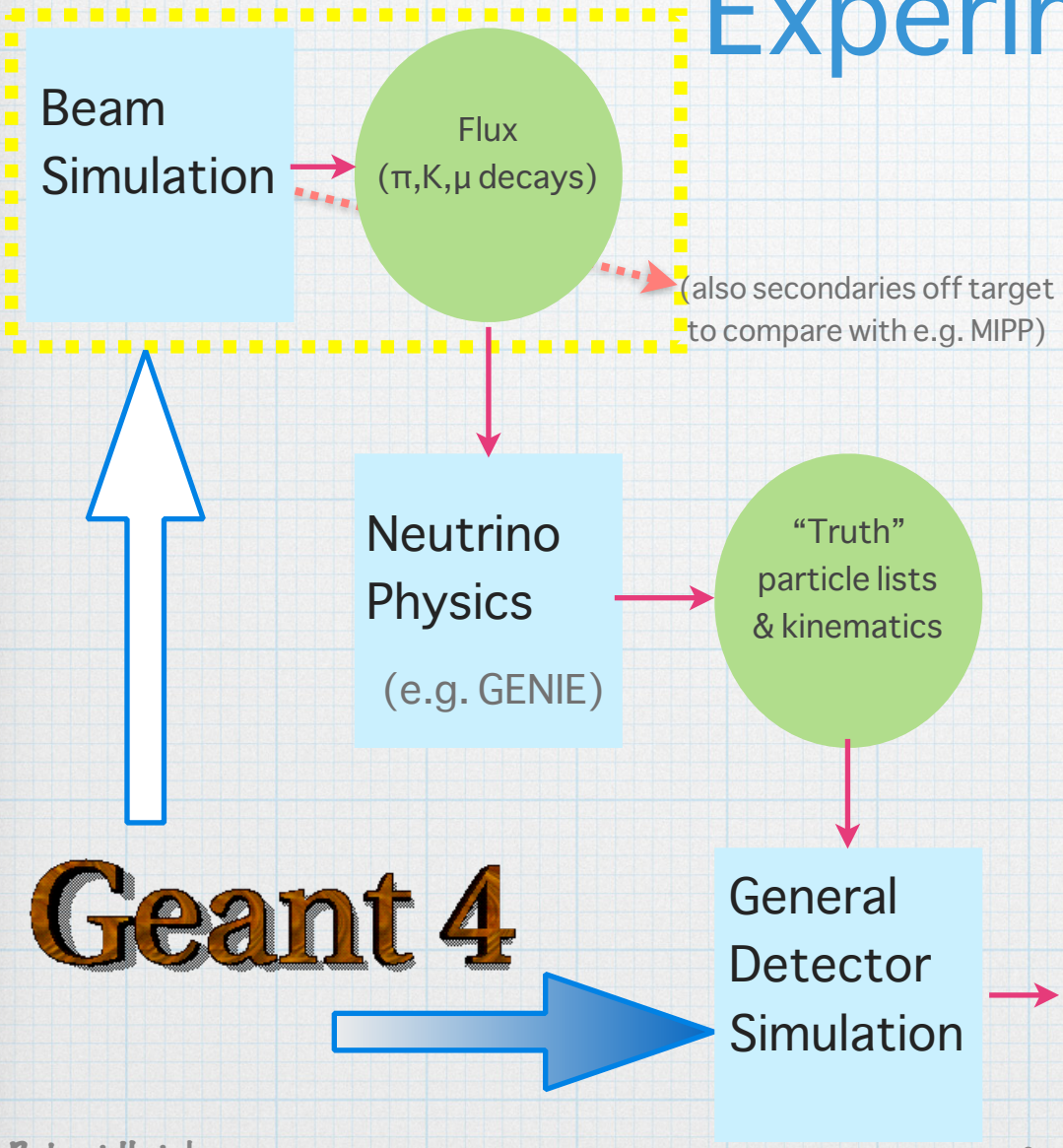
- Overview of neutrino simulations
- The art framework, LArSoft, NuTools
- GENIEHelper: Wrapping GENIE up
  - modifying GENIE behavior
  - brief mention of NuReweight
- LArG4: The Geant4 art module
  - Physics Lists
  - User Actions
  - Suggestions for moving forward
- Summary

General Simulation Workflow & Products in Neutrino Experiments



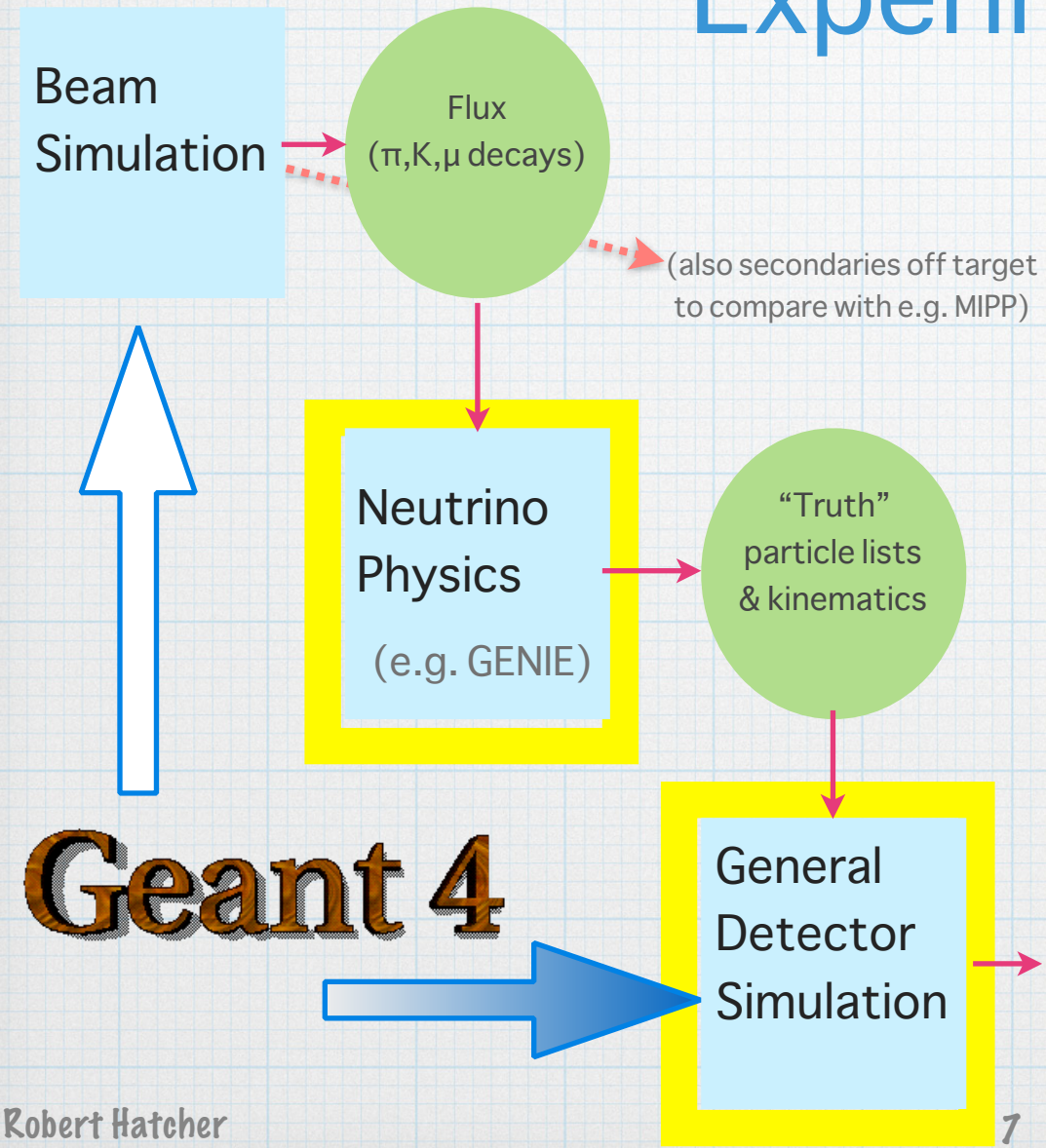
- We factorize the steps to make them tractable problems
  - Simulation of the beamline
  - Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

General Simulation Workflow & Products in Neutrino Experiments



- We factorize the steps to make them tractable problems
  - Simulation of the beamline
  - Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

General Simulation Workflow & Products in Neutrino Experiments



- We factorize the steps to make them tractable problems
  - Simulation of the beamline
  - Simulation of the detectors
- Different energy scales
- Even detector simulations have large variation in needs due to a variety of technology

Geant 4

LArSoft

DUNE Specific

LArIAT Specific

SBND Specific

User Code

LArPandora *pattern recog*

LArEvent Display

LArExamples

LArAna

LArReco

LArSim *(inc WireSim)*

NuTools *(LArSoft + NOvA)*

Geant4

GENIE

LArEvt *(calibration & space charge interfaces)*

LArData

LArCore *(inc Geometry)*

art Framework

Run/Subrun/Event Stores

Event Loop & Paths

MessageService

Configuration (FHiCL)

Metadata

Provenance Tracking

Dynamic Library Loading

I/O Handling (via ROOT)

boost

gccxml

SQLite

ROOT

libxml

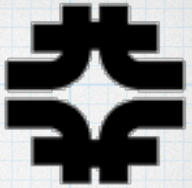
...

gsl

Geant 4

LArSoft Lines Of Code

circa 2016-04-22



```
$ cloc /grid/fermiapp/products/larsoft/larcore/v05_00_02/source/ /grid/fermiapp/products/
larsoft/lardata/v05_03_00/source/ /grid/fermiapp/products/larsoft/nutools/v1_23_02/
source /grid/fermiapp/products/larsoft/larsim/v05_02_00/source/ /grid/fermiapp/products/
larsoft/larreco/v05_03_00/source/ /grid/fermiapp/products/larsoft/larana/v05_03_00/source/
/grid/fermiapp/products/larsoft/larpandora/v05_02_01/source/ /grid/fermiapp/products/
larsoft/lareventdisplay/v05_02_00/source/ /grid/fermiapp/products/larsoft/larexamples/
v05_00_07/source/
    1310 text files.
    1308 unique files.
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                            701          49671          43680         205556
C/C++ Header                   570          15521          26100          35249
XML                              8             41             72            757
Bourne Shell                     8             70             78            142
Perl                             1             19              6             74
-------------------------------------------------------------------------------
SUM:                          1288          65322          69936         241778
-------------------------------------------------------------------------------
```

241,778

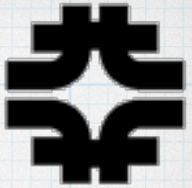
```
$ cloc /grid/fermiapp/products/larsoft/nutools/v1_23_02/source
       /grid/fermiapp/products/larsoft/larsim/v05_02_00/source/
     306 text files.
     305 unique files.
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                            162           9496           8762          44359
C/C++ Header                   131           2379           3573           6695
XML                              2              8             18            116
Perl                             1             19              6             74
Bourne Shell                     3             52             63             70
-------------------------------------------------------------------------------
SUM:                           299          11954          12422          51314
-------------------------------------------------------------------------------
```

51,314

Simulation ~ 21% of total LOC

includes connection to GENIE & Geant4, G4 User Actions, as well as WireSim

Geant 4



art / root Lines Of Code

```
$ cloc [top of the head of the art repository, not included the 'doc' directory]

   1436 text files.   1366 unique files.    497 files ignored.
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                            377           6040           4174          36555
C/C++ Header                   367           6781           7366          23649
CMake                           72            649            679           4042
Bourne Shell                    78            504            251           1365
Perl                            30            267             97           1210
C                                2             96            570           1204
Bourne Again Shell               7             53             39            356
Python                           1             13             32            230
XML                             10             20             30            177
C Shell                          1              3              5             45
MXML                             1              1              0              4
sed                              1              0              0              1
-------------------------------------------------------------------------------
SUM:                           947          14427          13243          68838
-------------------------------------------------------------------------------
```

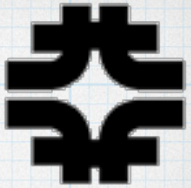
68,838

```
$ ~/bin/cloc root

   10966 text files.   10751 unique files.    1810 files ignored.
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C++                           3869         253948         353006        1257809
C/C++ Header                  3925          93108         128715         410475
C                              338          30144          33047         180459
Bourne Shell                    98           2686           3406          25895
Objective C++                   80           5366           4079          18575
HTML                           192           2206            159          16633
JavaScript                      18           2602            994          13454
Fortran 77                       4            351            540          12570
[others file types]
-------------------------------------------------------------------------------
SUM:                          9163         397933         530773        1979673
-------------------------------------------------------------------------------
```

1,979,673

Code Repo Overview



NuTools

NuTools: shared code for LArSoft + NOvA + other art-based ν expts.

```
EventDisplayBase
EventGeneratorBase
    CRY
    GENIE
        GENIEHelper.h
    GiBUU
G4Base
    ConvertMCTruthToG4.h
    DetectorConstruction.h
    ExampleAction.h
    G4Helper.h
    PrimaryParticleInformation.h
    UserAction.h
    UserActionFactory.h
    UserActionManager.h
IFDatabase
MagneticField
NuBeamWeights
NuReweight
    GENIEReweight.h
    ReweightLabels.h
    art
        NuReweight.h
        ReweightAna_module.cc
SimulationBase
    GTruth.h
    MCFlux.h
    MCNeutrino.h
    MCParticle.h
    MCTrajectory.h
    MCTruth.h
```

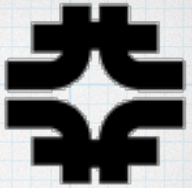
undergoing re-factorization data objects vs art module code

LArSim

Purview of LArSoft collaboration

```
DetSim      SimWire_module.cc
EventGenerator
    CORSIKA
    CRY
    GENIE       GENIEGen_module.cc
    MuonPropagation
LArG4       LArG4_module.cc
MCCheater
MCDumpers
MCSTReco
PhotonPropagation
RandomUtils
SimFilters
Simulation
    AuxDetSimChannel
    LArG4Parameters
    LArVoxel*
    ParticleHistory
    ParticleList
    SimChannel
    SimPhotons
    <other stuff>
TriggerAlgo
```

Event Generation w/ GENIE



- NuTools evgb::GENIEHelper wraps up GENIE for art
  - lots of configurability {GENIE event types; top volume; flux …}
    - https://cdcvs.fnal.gov/redmine/projects/nutools/wiki/GENIEHelper
      - sub-pages have additional information
  - Sample() fills { simb::MCTruth, simb::GTruth, simb::MCFlux }
    - call repeatedly (per art record) until "pile-up" condition is met
  - API also provides:
    - ctor: initialize w/ pset + geometry (TGeoManager, filename, "mass")
    - end-of-run: TotalExposure()
- LArSim evgen::GENIEGen_module
  - fetch geometry (ROOT TGeoManager, filename, "mass"); passes fhicl::ParameterSet to GENIEHelper
    - initialize pset from alternative random # seed service
  - produce() accumulates std::vector of simb::MCTruth, simb::GTruth, simb::MCFlux (+ art::Assns) and puts it in the art::Event
    - add "PassEmptySpills" parameter beyond GENIEHelper
    - fill sumdata::RunData, sumdata::POTSummary, sim::BeamGateInfo
  - create/fill lots of histograms

Geant 4

GENIEHelper - NuTools - F... | Customize Firefox

https://cdcvs.fnal.gov/redmine/projects/nutools/wiki/GENIEHelper

Search

Home My page Projects Help

NuTools

Search:

Overview Activity Issues New issue News Documents Wiki Files Repository Settings

NuTools » EventGeneratorBase »

Edit Watch Lock Rename Delete

GENIEHelper

GENIEHelper is the ART glue to GENIE. This section of the wiki is intended to help document using the `GENIEHelper` class (and `.fcl` file) to configure GENIE to individual needs.

Quick Guide to FHCL parameter variables

Basic GENIEHelper Parameters

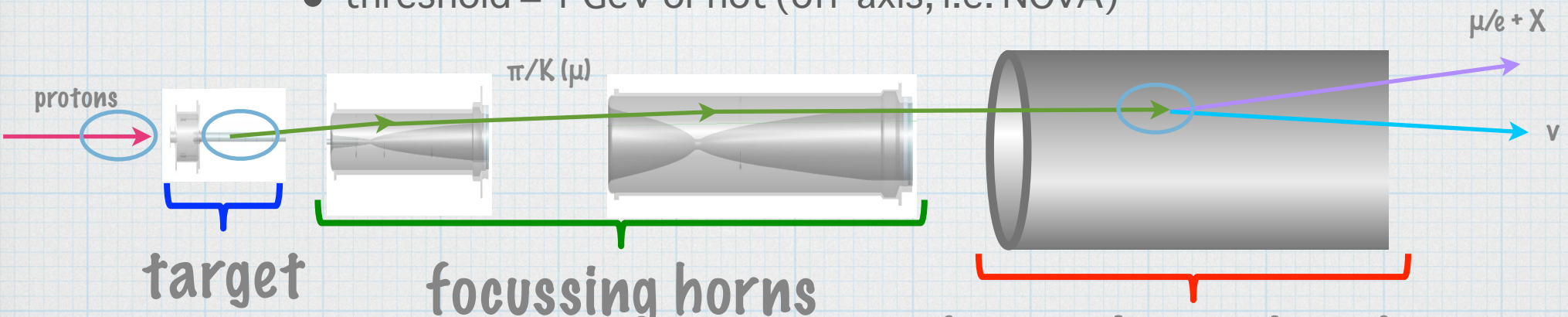
Name	Description
	single interactions or pile-up?
`POTPerSpill`	pile-up parameter (either this or the next should be 0)
`EventsPerSpill`	pile-up parameter - generate fixed # of GENIE events per ART record

Robert Hatcher

Into the weeds with GENIE

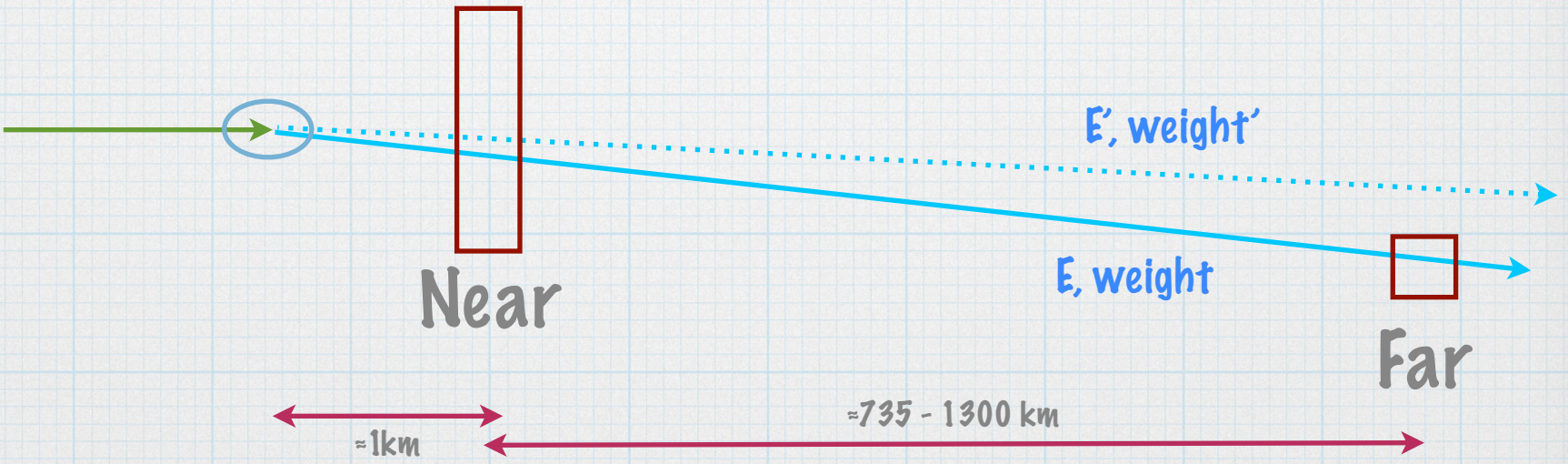
Beam Simulation

- Geant4 geometry (C++) + [G4 or Fluka physics (fortran)]
  - G4 geometry is quite detailed (and good match to as-built)
  - [fluka, if used, physics everywhere (not just target)]
- Record decay, initial secondary production and initial proton info
  - now: ancestors history from initial proton to decay products
  - 2ndary production models are active areas of study
- Possibly uses importance weights and thresholds
  - $w_\pi = \min(\max(30/p_{tot}, 1) * w_{parent}, 100)$
  - threshold = 1 GeV or not (off-axis, i.e. NOvA)



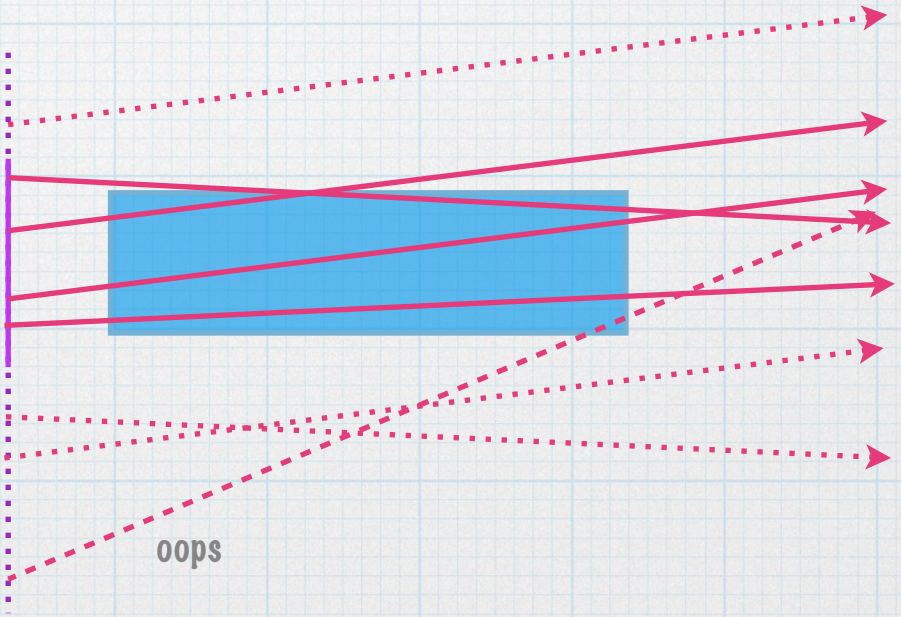
Decay Reweighting

- The probability that a decay results in a neutrino ray that goes through any point depends on the relativistic boost at the decay point; the ν energy will also depend on position
- Near and Far detectors subtend a different angular size → they see different spectra



Flux Window

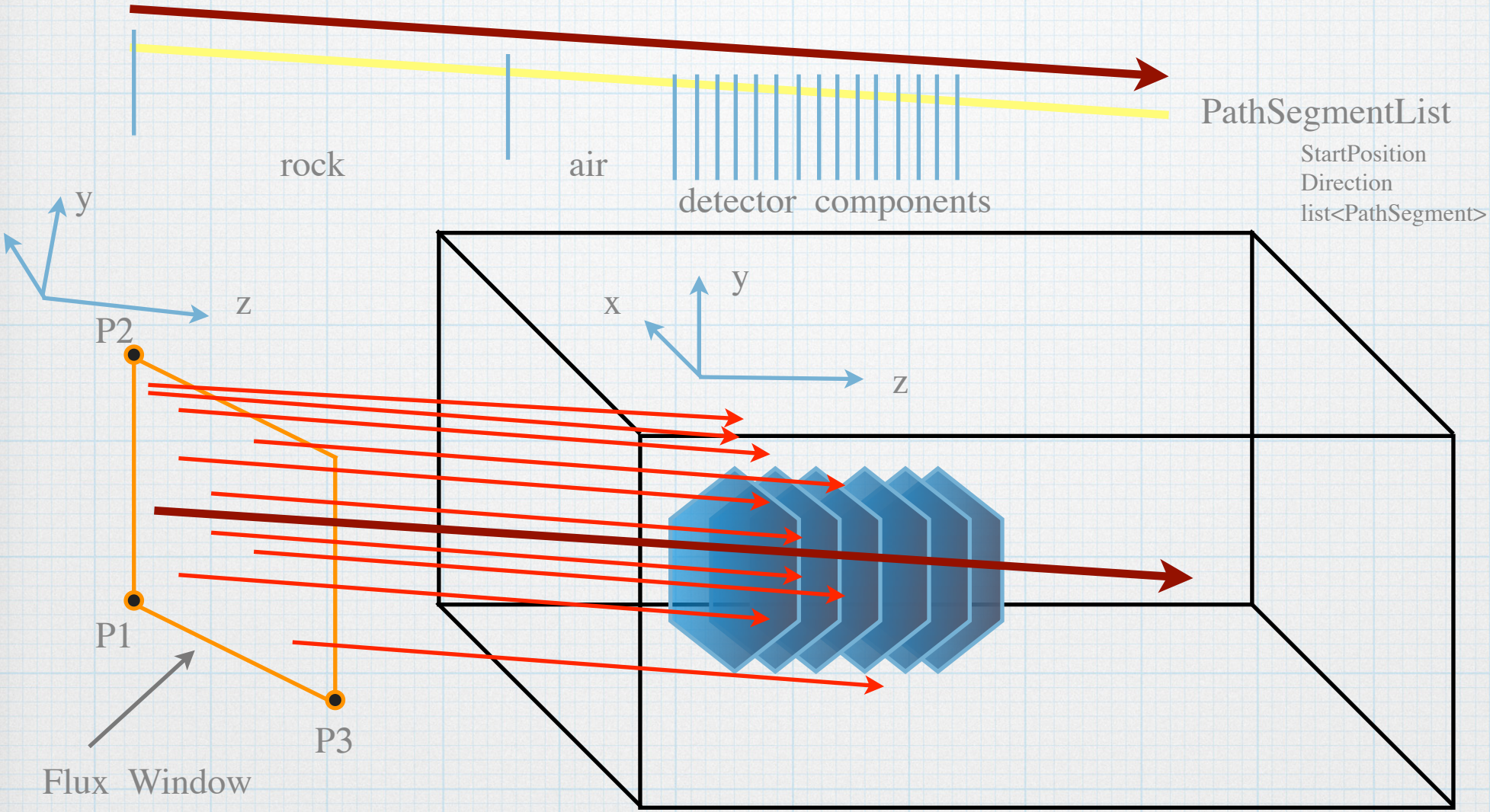
- A fictitious parallelogram in space from whence neutrino rays emanate
- needs to be sized:
  - large enough that all (to best approximation) relevant rays that might run through the geometry pass through the window
  - small enough to exclude rays that aren't of interest



GNuMIFlux/GDk2Nu vs GSimpleNtpFlux

- GENIE's GNuMIFlux or GDk2Nu
  - read an entry from ntuple of decay info
  - pick random point on flux window (x,y,z)
  - calculate x-y weight, energy, p4
  - accept/reject based on weights ($wgt_{x\text{-}y} * wgt_{importance}$)
  - (possibly) push backwards along ray to (x',y',z0)
  - ⇒PathSegmentList created from this ray
  - cycling back to same entry won't give same ray
    - different window point ⇒ different weight, energy, trajectory
- GENIE's GSimpleNtpFlux
  - simple ntuple format of flavor, position, direction, weight
    - provision for carrying extra info to allow limited hadron reweighting
  - some file level meta data (window position, total protons,...)

ν Rays and Geometry



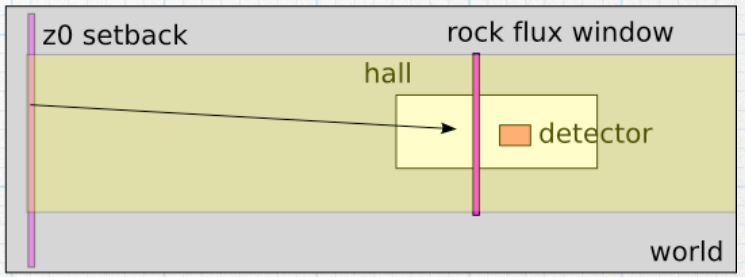
GENIE x-sections

- GENIE cross-sections distributed as UPS product
  - UPS sets `$GENIEXSECFILE` (`$GENIEXSECPATH/gxspl-FNALsmall.xml.gz`)
  - details found in `$GENIEXSECPATH/README`
  - not differential x-sec; only as a function of neutrino energy
  - 500 knots, [0.01:400] GeV spaced logarithmically
    - 5% of points > 120; flux has long E_ν tail NOvA MN flux has entries up to 104 GeV
    - could study effect of fewer knots - accuracy vs. size
  - file size ~750 MB (23584 splines)
  - 272 for proton, 302 for neutron, 590 for each of 41 nuclei
    - all 6 ν flavors
    - e.g. C^{12}, N^{14}, O^{16}, Na^{23}, Al^{27}, Si^{28}, Cl^{35}, Ar^{40}, K^{39}, Ti^{48}, Fe^{56}, Ba^{137}, ...
    - `gxspl-FNALlarge.xml.gz` has 106 isotopes (needs unpacking)
  - For non-standard configurations UPS also distributes `UserPhysicsOptions.xml` and/or `EventGeneratorListAssembler.xml`
    - add path to `$GXMLPATH` so GENIE finds it

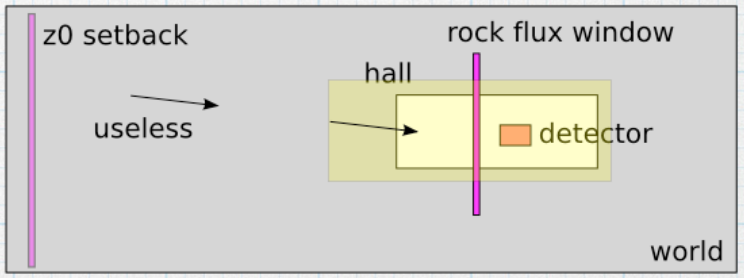
Choosing a Vertex "Outside the Box"

- When a "topvol" isn't set, GENIE considers the entire geometry
- GeomSelectorRockBox trims the volume to the hall + minimum safety + a size proportional to the neutrino energy

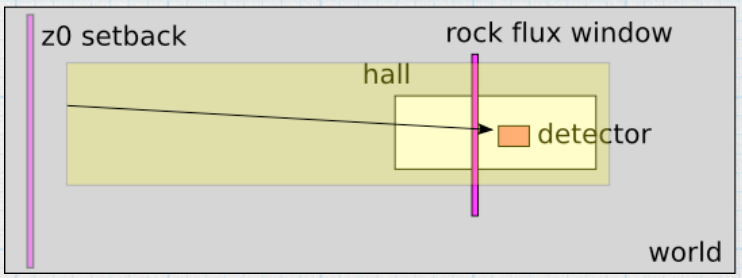
No GeomSelector



RockBox: 2 GeV

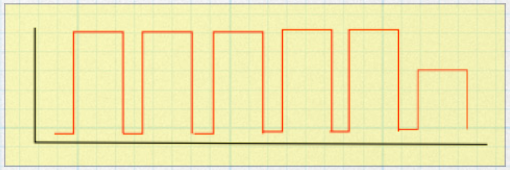


RockBox: 80 GeV



Overlay Pile-up

- Collect events
  - MINOS: pull from input sample files
    - Poisson distribution: nDetPerSpill + nRockPerSnarl for a given intensity
    - single use of detector events, randomize pulling from rock files (reuse, except once)
  - NOvA: generate events until used fixed POTs/Spill
- Distribute events in time over spill interval according to intensity profile

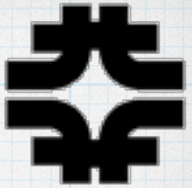


  - offset truth info times (StdHep/HepMC)
  - also offset corresponding hit times, if already propagated in GEANT
  - if combined particle list, adjust parentage indices
  - add any event kinematics/flux records to list for spill
    - good to have mechanism tying kin/flux to particle list

Why Can't I … ?

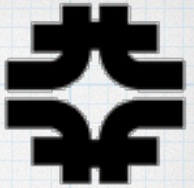
- "Okay, I've looked at the GENIEHelper wiki and I don't see any way to change physics parameters (e.g. M_A, MEC values) or choose physics models from within a FHiCL file for event generation. What gives?"
  - short answer: just because …you don't want the wrong answer…
  - long answer: GENIE must be run in a consistent mode
    - to decide whether a ν flux ray interacts at all and to pick event vertex GENIE samples the material along the ray's path for the amount of material transversed
    - $P_{interaction} \simeq$ the number of nuclei × total cross-section
    - total cross-section splines are pre-calculated
      - this is computationally expensive!
    - controlled by GENIE `UserPhysicsOptions.xml` (and possibly `EventGeneratorListAssembler.xml`)

Why is it … so slow?



- Start up
  - spline reading
    - Loading from ROOT file (vs XML) speeds up loading, but doesn't appear to significantly change peak memory usage
  - geometry exploration
    - Use pre-calculated max path lengths
  - flux handling
    - limit number of files copied to local disk `MaxFluxFileMB`
- Event Generation
  - off-axis flux w/ Dk2Nu & GNuMI ?
    - Use pre-transformed flux `GSimpleNtpFlux`
  - atmospheric flux
    - GENIE has ideas about how to improve this; lack manpower
  - rock event generation
    - tuning … (also, is it the GENIE stage or G4 stage?)

Why is the memory footprint big?



- Total Cross-Section Spline File `gxspl-FNALsmall.xml`
  - many knots (500 over E range of [0.01:400] GeV)
    - probably not due to energy range (only a few knots above beam E)
    - need a study "smoothness" vs. # of knots
  - many isotopes (not all needed for all detector geometries)
    - single XML file … only read once, no provision for multiple files
    - teach GENIE how to do lazy loading … on the to-do list 😬
    - also adding unnecessary isotopes hurts everyone
      - if not in x-sec spline file: GENIE generation gives up (needed to set max prob)
      - e.g. Beryllium was added to wire composition
        - ~2% of wire mass is Beryllium (added to originally pure Copper)
        - wires are ~0.0023% of the LAr mass in DUNE FD

What Else?

- NuReweight package - interface to GENIE functionality
  - new weights can be calculated for existing GENIE events by changing physics parameters (e.g. M_A) and (some) models.
    - need to reconstitute `genie::EventRecord`
    - ratio of differential cross-section
    - doesn't change reco interference from "bkg" events
  - used to vary models in order to study systematics
  - adjust existing MC samples to new central values w/out regeneration
  - needs someone to take primary lead for maintenance
    - original author has left the field
    - reported issue w/ speed … might need restructuring

GENIE Questions?

- A brief pause …
- Ask now or forever hold your peace …
- No, not really … ask me any time

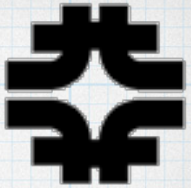
Tools for Geant4 Use

- NuTools' G4Base provides help, but less comprehensive
  - Different from generators (e.g. GENIE) for which a unified representation of "interactions" (MCTruth) exists; but that doesn't hold for output products of Geant4 — energy losses, # photons …
  - Serves as a tool kit (much like G4 itself)
    - https://cdcvs.fnal.gov/redmine/projects/nutools/wiki/G4Base
  - `G4Helper` - basic setup in `art` framework of G4 fundamentals
  - `DetectorConstruction` - create G4 geometry from GDML file
  - `ConvertMCTruthToG4` - "does what it says on the tin"
  - `G4PhysListFactory` (alternative to official standard G4) - choose G4 physics lists at run time
  - `UserAction [ Factory ]` - user, ah, actions at various points in G4 running
    - see followup pages
- LArSim
  - `larg4::LArG4_module`
  - register own `larg4::PhysicsList` for use w/ `G4PhysListFactory`
  - other LAr specific code

Let's follow this guy…what could go wrong?

Geant 4

```
53
54  namespace g4b {
55
56    class UserAction {
57
58    public:
59
60      UserAction() {};
61      UserAction(fhicl::ParameterSet const& pset) { Config(pset); }
62      virtual ~UserAction() {};
63
64      /// Override Config() to extract any necessary parameters
65      virtual void Config(fhicl::ParameterSet const& /* pset */ ) {};
66
67      /// Override PrintConfig() to print out current configuration
68      virtual void PrintConfig(std::string const& /* opt */ ) {};
69
70      /// The following a list of methods that correspond to the available
71      /// user action classes in Geant 4.0.1 and higher.
72
73      /// G4UserRunAction interfaces
74      virtual void BeginOfRunAction  (const G4Run*  ) {};
75      virtual void EndOfRunAction    (const G4Run*  ) {};
76
77      /// G4UserEventAction interfaces
78      virtual void BeginOfEventAction(const G4Event*) {};
79      virtual void EndOfEventAction  (const G4Event*) {};
80
81      /// G4UserTrackingAction interfaces
82      virtual void PreTrackingAction (const G4Track*) {};
83      virtual void PostTrackingAction(const G4Track*) {};
84
85      /// G4UserSteppingAction interface
86      virtual void SteppingAction    (const G4Step* ) {};
87
88      /// Does this UserAction do stacking?
89      /// Override to return "true" if the following interfaces are implemented
90      virtual bool ProvidesStacking() { return false; }
91      /// G4UserStackingAction interfaces
92      virtual G4ClassificationOfNewTrack
93        StackClassifyNewTrack(const G4Track*) { return fUrgent; }
94      virtual void StackNewStage() {};
95      virtual void StackPrepareNewEvent() {};
96
97      // allow self-identification
98      std::string const & GetName() const { return myName; }
99      void                SetName(std::string const& name) { myName = name; }
100   private:
101     std::string myName;  ///< self-knowledge
102   };
103
104 } // namespace g4b
105
106 #endif // G4BASE_UserAction_H
107
```



here “run”
means “event”

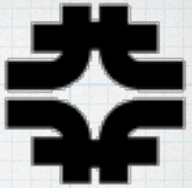
😬

each `MCTruth`
is a separate

```
fUIManager->ApplyCommand(
"/run/beamOn 1");
```

Robert Ha

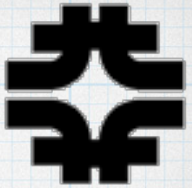
larg4::LArG4_module



- Enables fhicl option for geometry overlap checking
  - could extend to also turn off GDML Schema validation (G4Base option)
- Written so as to only add larg4::ParticleListAction
  - in principle one could, as NOvA does, allow users to append additional UserAction derived classes.
    - Why? something like g4n::RockCutterAction
- produce() does
  - extract std::vector< art::Handle< std::vector<simb::MCTruth> > > from given labels, or just everything it can find
  - feeds each to G4 with fG4Help->G4Run(aMCTruth)
  - for each, get back std::vector<simb::MCParticle> , make Assn to MCTruth
  - at end, puts into the art::Event, those +
    - std::vector<sim::SimChannel>
    - std::vector<sim::SimPhotons[Lite]>
    - std::vector<sim::AuxDetSimChannel>

could do with some trimming of #includes

Sounds innocuous so far …



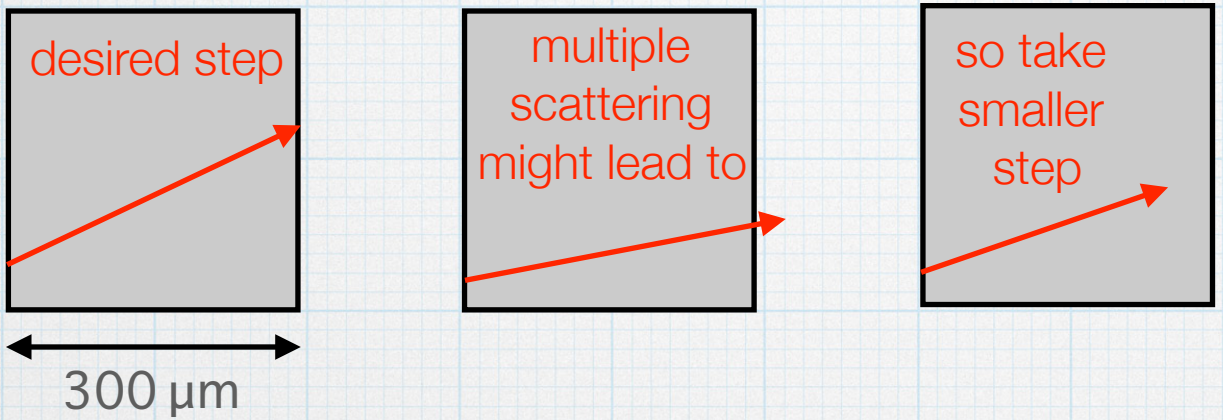
- Some possible issues hidden in LArSim `larg4::PhysicsList`
  - non-standard; needs maintenance w/ G4 version changes
    - limits options: probably some things in LArSim don't work if one chooses a pre-defined G4PhysList from G4 base release
  - dual parallel world geometries (in addition to the base geometry)
  - singletons … decouples code … but thread safe headache down the road

Ugh

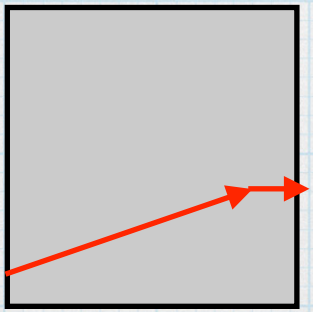
```
// Note that the name below MUST match the name used in the
// LArVoxelReadoutGeometry or OpDetReadoutGeometry constructor.
LArVoxelParallelWorldScoringProcess->SetParallelWorld("LArVoxelReadoutGeometry");
OpDetParallelWorldScoringProcess->SetParallelWorld("OpDetReadoutGeometry");
```

Parallel Geometries

- Force G4 to consider all boundaries during transport
  - as I understand it, LAr is "voxelized" into $(300\ \mu m)^3$ cubes [default]
  - thought to be there to limit step size, force localized e/γ depositions
    - if so, this is probably not the optimal means of doing this



Zeno's … paradox… ensues



eventually land on boundary surface

- In the mean time:
  - odd step structure
  - lots of overhead in transportation process
  - increased memory footprint
  - makes geometry unvisualizable
  - harder to validate geometry

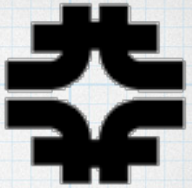
Alternative to Parallel Geometries

- If this really is the reason, then probably an misuse of G4
- Two possibilities for how to achieve the same effect:
  - add physics process that limits the step size
  - special tracking cuts
    - limit step size for particular particles in particular volumes
    - http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/BackupVersions/V10.1/html/ch05s04.html#sect.ProThres.Spe

Other Geant4 Issues

- Trade off in physics fidelity vs. speed for EM options
  - needs exploration
- One of the physics processes (Scintillation?) is a fork of a standard G4 class just to expose interface that reports number of photons, rather than stack/toss entries
  - latest G4 tag includes this interface (or equivalent)
  - this class can then be removed when that G4 version is adopted
    - improves maintainability … makes it someone else's job.
- Is radioactive decay of interest? … add to physics list?

Testing, Testing… Is this thing on?



- By “upstream” vendors …
  - GENIE: release validation (improving…big push this summer)
  - G4: release validation [Julia] - monitor “standard” setups
  - G4: physics/performance [Hans on LArIAT]
    - possibly propose to G4 to some tests targeted toward LAr concerns
    - similar to “simplified CMS” test
- By LArSoft Expt / Users …
  - none of the above is guaranteed to cover regions of interest to Expt X
    - different mix of particles, energies, materials (okay, here mostly LAr, but few of the above tests are on LAr), physics processes, resolutions/ sensitivities/thresholds
  - need tests that
    - ensure LArSim/NuTools hasn’t broken interface for a fixed G4 version
    - validate physics quantities when G4 itself changes

Backup Slides

ν at the Intensity Frontier

Tertiary Test Beams

Fermilab LArIAT ‡ ✅

CERN DUNE proto-types
single & dual phase ✅ ✅

Neutrino Beams

present & future (and recent past)

- NuMI (Main Injector)
  - LE & ME target/horn configurations
- Booster Neutrino Beam
- LBNF under design

Neutrino Detectors

MINOS [+] ‡ (Near & Far detectors - magnetized)

ArgoNeuT † ✅
(same small LAr detector in test beam / NuMI beam)

MINERvA ‡ (fine grained & multi-target material)

NOvA ‡ (Near & Far detectors - off-axis)

SBND ✅
(Short Baseline Near Detector Expt, formerly LAr1ND)

ANNIE
(to study neutron production in water using BNB ν)

μBooNE ‡ ✅

miniBooNE †

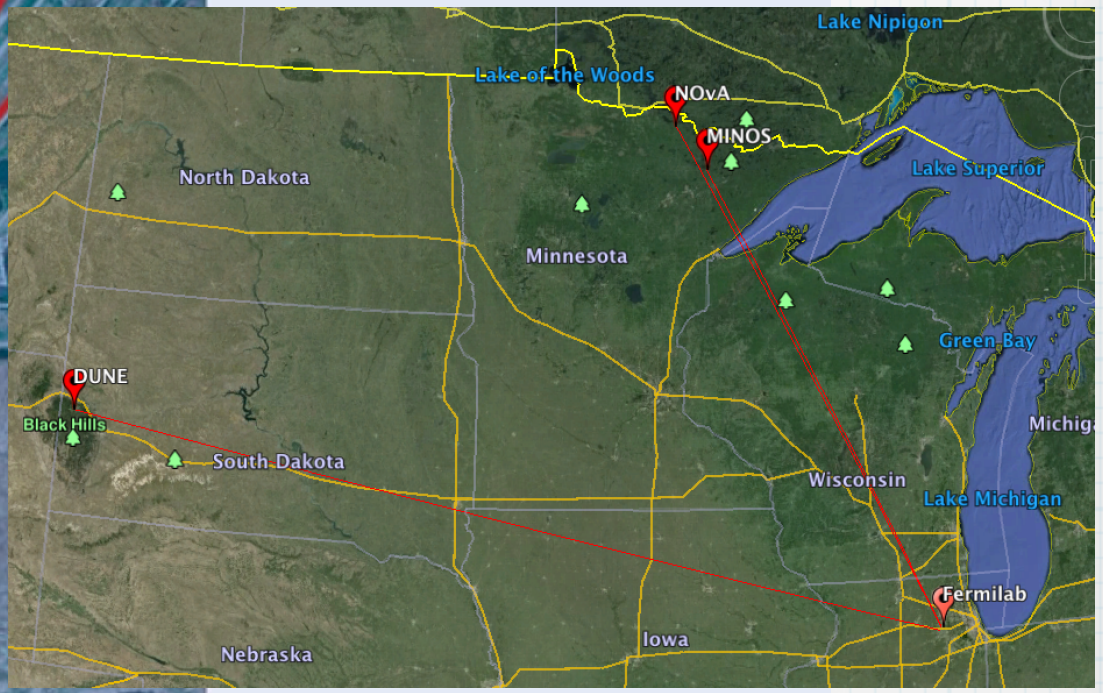
ICARUS-T600
(to be refurbished & moved from Gran Sasso National Lab in Italy to serve as BNB Far Detector)

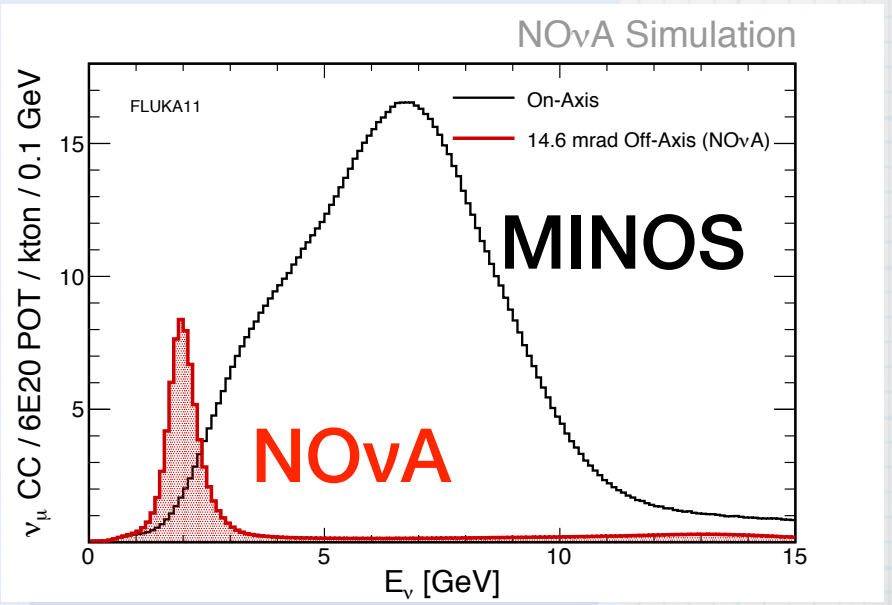
DUNE ✅+ ✅
(Deep Underground Neutrino Experiment, formerly LBNE)
(Near & Far detectors + 35 ton + test beam single & dual phase prototypes at CERN)

† ran previously
‡ currently running

✅ LArTPC
(Liquid Argon TPC)
Technology


to Minnesota
test beam
LArIAT
Wilson Hall
NuMI beam
MINOS, Minerva, NOvA
linac
booster
muon campus
μBooNE, SBND, ICARUS,
booster neutrino beam
to South Dakota
DUNE
muons
nuclei
kaons
Main Injector and Recycler
Project X
Tevatron Ring (decommissioned)


Lake Nipigon
Lake of the Woods
NOvA
MINOS
Lake Superior
North Dakota
Minnesota
Green Bay
DUNE
Black Hills
South Dakota
Wisconsin
Lake Michigan
Fermilab
Nebraska
Iowa


NOνA Simulation
FLUKA11
On-Axis
14.6 mrad Off-Axis (NOνA)
MINOS
NOνA
ν_μ CC / 6E20 POT / kton / 0.1 GeV
E_ν [GeV]
0
5
10
15

A bit about events: topologies

Events & Backgrounds:

- ν "detector" (LAr vs. cryostat),
- ν in surrounding "rock"/"dirt"
- cosmics (single μ, multi-particle)
- radiological sources
- spallation in the rock
- astrophysical (many very small energy depositions)
- nucleon decay ($p \rightarrow K^{+}\nu$ & $p \rightarrow K^{0}\mu^{+}$)

ν induced events:

- "pile-up" distributed in time (10μsec vs. drift time) & space (different from colliders)
- basic ν topologies:
  - CC nu-mu
  - CC nu-e
  - NC (inc pi0)

Much of the physics is in distinguishing these

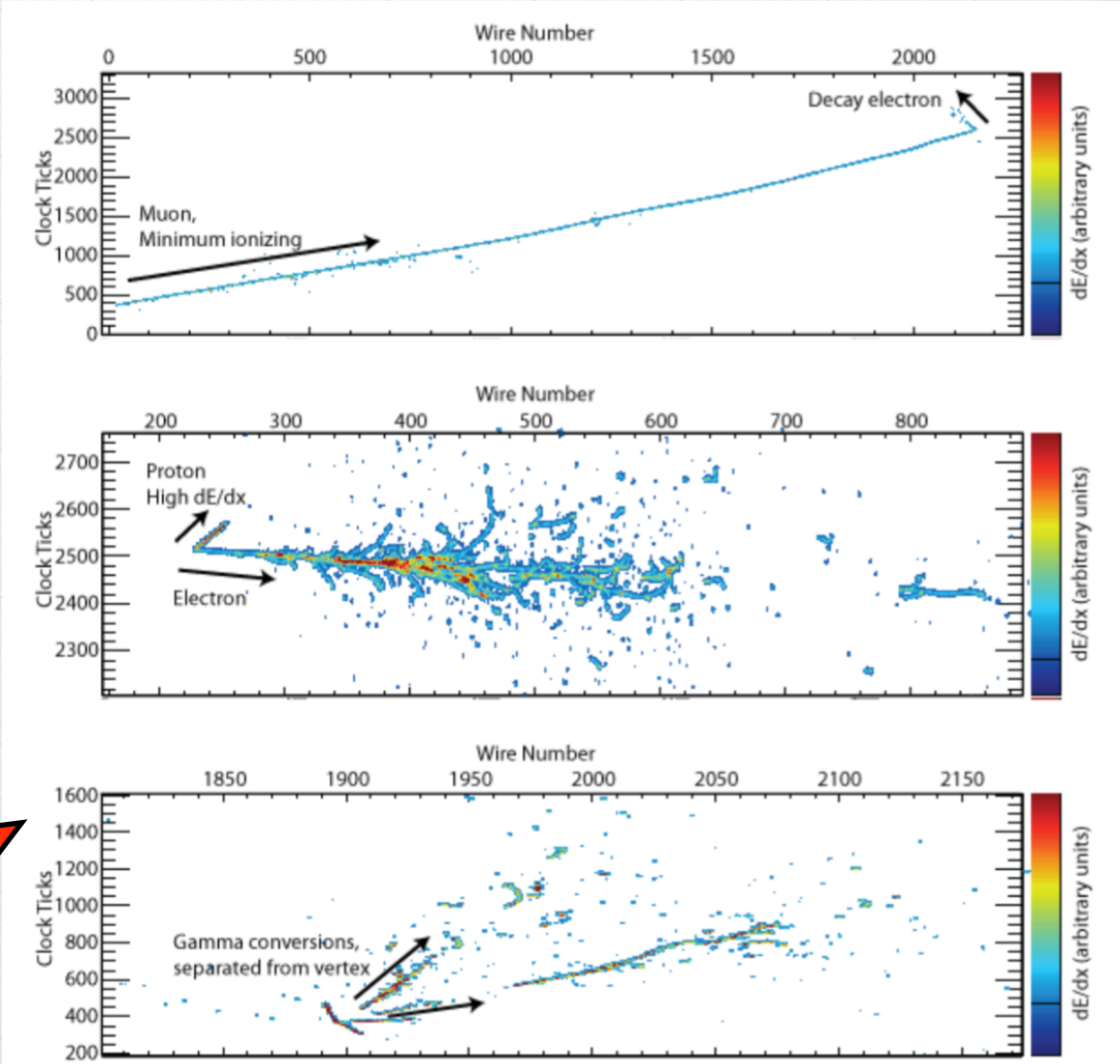
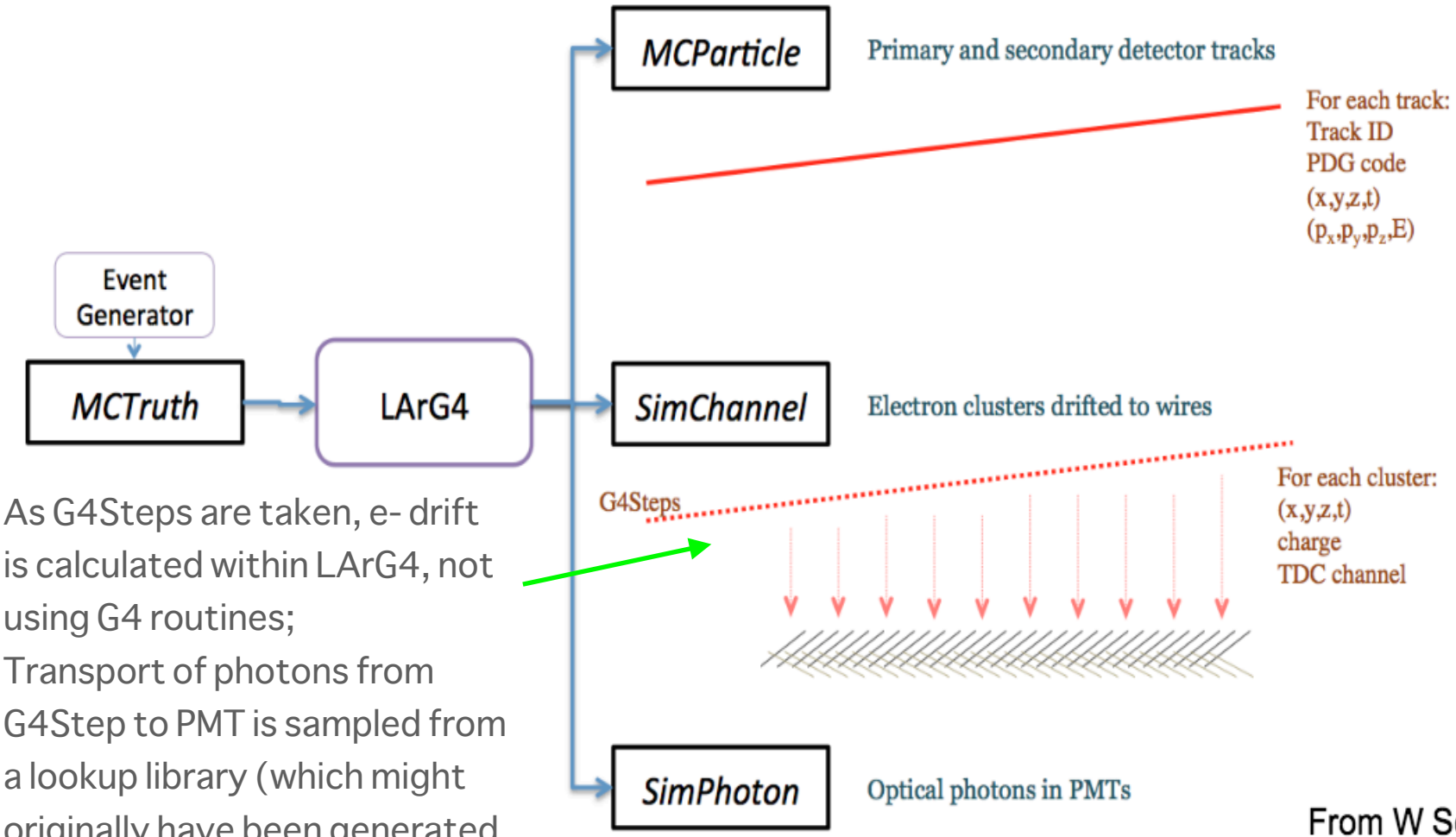


Figure A.1: Examples of neutrino beam interactions in a LArTPC obtained from a Geant4 simulation [220]. A ν_μ-CC interaction with a stopped μ followed by a decay Michel electron (top), a ν_e-CCQE interaction with a single electron and a proton (middle), and an NC interaction which produced a π^0 that then decayed into two γ's with separate conversion vertices (bottom).

Detector simulation



As G4Steps are taken, e- drift is calculated within LArG4, not using G4 routines;
Transport of photons from G4Step to PMT is sampled from a lookup library (which might originally have been generated with Geant4)

From W Seligman

Some Benchmarking Results

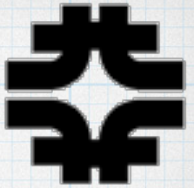
LArIATSoft: single charged particle (<E>=780 MeV, σ =700 MeV)

Timing: art timer 1000 events (except reco: 100, only π)

Module	Timing ($\pi^{\pm}$) sec/evt		T($e^{\pm}$)		T($\mu^{\pm}$)	
Geant4	0.140	17.8% (1.10%)	0.388	39.2%	0.088	14.0%
ROOT I/O	0.252	32.1% (1.97%)	0.245	24.7%	0.203	32.2%
DetSim (WireSim)	0.394	50.1% (3.08%)	0.358	36.1%	0.339	53.8%
Reco (89% TrackMaker)	~12	(93.85%)	—	—	—	—

Note: Geant4 will scale linearly with complexity of events; reconstruction due to combinatorics will not!

Running DUNE Simulation



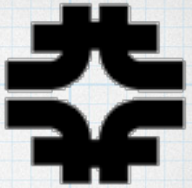
DuneTPC: 4 APA FarDet "workspace" (g4.9.6p03)

Timing: art timer (100 events)

Module	Timing	
GENIE Gen	0.1268 sec / evt †	0.05%
Geant4	3.842 sec / evt	1.55%
DetSim (WireSim)	44.1075 sec / evt	17.81%
Reco	199.4674 sec / evt	80.52%
MergeAna	0.1656 sec / evt	0.07%

Note: Geant4 will scale linearly with complexity of events; reconstruction due to combinatorics will not! † Not including x-sec load time (~80s)

The art framework



```
art v1_17_07  -q debug:e9:nu
|__cetpkgsupport v1_10_01 -g current
|__messagefacility v1_16_22  -q debug:e9
|  |__fhiclcpp v3_12_09  -q debug:e9
|     |__cetlib v1_15_04  -q debug:e9
|     |  |__cpp0x v1_04_13  -q debug:e9
|     |     |__boost v1_57_0a  -q debug:e9
|     |        |__gcc v4_9_3
|     |__sqlite v3_08_10_02
|__root v5_34_32  -q debug:e9:nu
|  |__clhep v2_2_0_8  -q debug:e9
|  |__fftw v3_3_4  -q debug
|  |__gsl v1_16a  -q debug
|  |__pythia v6_4_28d  -q debug:gcc493
|  |__postgresql v9_3_9  -q p2710
|  |  |__python v2_7_10
|  |__mysql_client v5_5_45  -q e9
|  |__xrootd v3_3_4d  -q debug:e9
|  |__libxml2 v2_9_2  -q debug
|__cppunit v1_12_1c  -q debug:e9
|__gccxml v0_9_20150423
|__tbb v4_4_0  -q debug:e9
```